

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**ANDRÉ BARBOZA DA SILVA**

**DESENVOLVIMENTO DE UM SISTEMA WEB PARA O CONTROLE DE  
TREINAMENTOS REALIZADOS POR COLABORADORES DE UMA EMPRESA**

**GUARAPUAVA**

**2025**

**ANDRÉ BARBOZA DA SILVA**

**DESENVOLVIMENTO DE UM SISTEMA WEB PARA O CONTROLE DE  
TREINAMENTOS REALIZADOS POR COLABORADORES DE UMA EMPRESA**

**DEVELOPMENT OF A WEB SYSTEM FOR THE CONTROL OF TRAININGS  
CONDUCTED BY EMPLOYEES OF A COMPANY**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Diego Marczal

Coorientadora Profª. Drª. Renata Luiza Stange

**GUARAPUAVA**

**2025**



[4.0 Internacional](#)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

## **AGRADECIMENTOS**

A Deus, minha gratidão maior. Sem Sua presença e direção, nada disso teria sido possível; reconheço que cada passo desta jornada foi sustentado por Sua graça e amor.

À pessoa que eu mais amo nesta vida, minha esposa e parceira, Evillyn Naiane. Obrigado por estar ao meu lado em todos os momentos. Sem você, nada disso teria o mesmo sentido. Sua companhia nos dias bons e ruins, seu apoio nas horas complicadas e a alegria que traz aos momentos simples fizeram toda a diferença para que eu chegassem até aqui.

Aos meus orientadores, Prof. Dr. Diego Marczal e Prof<sup>a</sup>. Dr<sup>a</sup>. Renata Luiza Stange, agradeço pela orientação dedicada, pelas discussões enriquecedoras e pelo apoio técnico e humano durante todas as etapas deste trabalho. A confiança e o apoio de vocês foram fundamentais.

A todos que, de alguma forma, direta ou indiretamente, contribuíram para a realização deste trabalho, deixo meu sincero agradecimento.

*“Tudo quanto te vier à mão para fazer, faze-o conforme as tuas forças, porque na sepultura, para onde tu vais, não há obra, nem projetos, nem conhecimento, nem sabedoria alguma.”*

*— Eclesiastes 9:10*

## RESUMO

O crescente investimento em capacitação profissional e a necessidade de garantir ambientes de trabalho mais seguros evidenciam a importância do controle adequado dos treinamentos realizados por colaboradores nas empresas. Em muitos casos, processos baseados em planilhas e documentos físicos tornam-se insuficientes para assegurar organização, rastreabilidade e acesso rápido às informações, especialmente quando há necessidade de comprovação de qualificações ou análise do histórico de capacitações. Diante desse cenário, este trabalho apresenta o desenvolvimento do Training System, um sistema web voltado ao registro, acompanhamento e gerenciamento de treinamentos corporativos. O objetivo central do sistema é oferecer um meio estruturado para armazenar informações, automatizar processos repetitivos e facilitar a identificação do estado dos treinamentos atribuídos aos colaboradores. A metodologia empregada envolveu o levantamento de requisitos, uso do método MoSCoW para priorização, modelagem de banco de dados, definição de fluxos de uso e implementação baseada no framework Ruby on Rails. O desenvolvimento contemplou funcionalidades essenciais, como autenticação, gerenciamento de usuários, setores e treinamentos, automação do vínculo de capacitações obrigatórias e controle de status dos treinamentos, com registro das datas relevantes para auditoria e acompanhamento. Os resultados obtidos demonstram que o sistema proporcionou uma estrutura mais clara para o gerenciamento das informações, permitindo centralização dos registros, redução de falhas comuns no uso de planilhas e maior facilidade no acesso aos dados. Assim, o Training System representa uma alternativa funcional para apoiar empresas na organização de seus processos de capacitação, contribuindo para maior confiabilidade e eficiência no controle de treinamentos corporativos.

**Palavras-chave:** sistema web; gerenciamento de treinamentos; capacitação corporativa; controle de processos; treinamento empresarial.

## ABSTRACT

The growing demand for professional training and the need to ensure safer and more efficient work environments highlight the importance of properly managing employee training records within companies. In many situations, processes based on spreadsheets and physical documents prove insufficient for ensuring organization, traceability, and quick access to information, especially when verifying qualifications or reviewing training histories. In this context, this work presents the development of the Training System, a web application designed to register, track, and manage corporate training activities. The main objective of the system is to provide a structured environment for storing information, automating repetitive processes, and facilitating the identification of the status of trainings assigned to employees. The methodology used included requirement analysis, prioritization through the MoSCoW method, database modeling, definition of system flows, and implementation with the Ruby on Rails framework. The system includes core functionalities such as authentication, user, department, and training management, automatic assignment of mandatory trainings, and status tracking with relevant date records for auditing and monitoring. The results indicate that the system offers a more organized structure for managing training information, enabling centralization of records, reducing common errors associated with spreadsheets, and improving the accessibility of stored data. Thus, the Training System functions as a practical solution for supporting companies in managing their training processes, contributing to greater reliability and efficiency in corporate training control.

**Keywords:** web system; training management; corporate training; process control; employee training.

## LISTA DE FIGURAS

<b>Figura 1 – Tela de dashboard do Training System com visualização por colaborador.</b>	12
<b>Figura 2 – Fluxo de trabalho do GitFlow. Fonte: (PX, 2023).</b>	15
<b>Figura 3 – Estrutura do método MoSCoW. Fonte: (SEBRAE, 2022).</b>	16
<b>Figura 4 – Diagrama de Atividades do Fluxo Principal do Sistema.</b>	17
<b>Figura 5 – Listagem dos Treinamentos Colaboradores: visualização dos registros de treinamentos dos usuários. Fonte: Autoria própria.</b>	22
<b>Figura 6 – Modelo Relacional do Banco de Dados. Fonte: Gerado com dbdiagram.io.</b>	23
<b>Figura 7 – Tela de gerenciamento de setores de um colaborador.</b>	30
<b>Figura 8 – Tela de detalhes do treinamento pendente para o colaborador.</b>	31
<b>Figura 9 – Tela de detalhes do treinamento em andamento para o colaborador.</b>	33
<b>Figura 10 – Tela de login.</b>	34
<b>Figura 11 – Diagrama de Atividades - Fluxo do Administrador.</b>	35
<b>Figura 12 – Tela inicial do administrador.</b>	36
<b>Figura 13 – Tela de cadastro de setor.</b>	36
<b>Figura 14 – Tela de cadastro de colaborador.</b>	37
<b>Figura 15 – Tela de vínculo de colaborador a setor.</b>	37
<b>Figura 16 – Tela de cadastro de treinamento.</b>	38
<b>Figura 17 – Tela de monitoramento de status (Visão do Administrador).</b>	38
<b>Figura 18 – Diagrama de Atividades - Fluxo do Colaborador.</b>	39
<b>Figura 19 – Tela de listagem de treinamentos do colaborador.</b>	40
<b>Figura 20 – Tela de detalhes do treinamento (em andamento) para o colaborador.</b>	40
<b>Figura 21 – Tela de perfil do colaborador.</b>	41
<b>Figura 22 – Tela de Login: interface de autenticação do usuário. Fonte: Autoria própria.</b>	54
<b>Figura 23 – Listagem de Treinamentos: visualização dos treinamentos disponíveis. Fonte: Autoria própria.</b>	55
<b>Figura 24 – Opções de Conta: configurações de perfil e ajustes pessoais. Fonte: Autoria própria.</b>	55
<b>Figura 25 – Tela Inicial do Administrador: painel principal de acesso e navegação. Fonte: Autoria própria.</b>	56

<b>Figura 26 – Cadastro de Treinamentos: formulário para inclusão de novos treinamentos. Fonte: Autoria própria. . . . .</b>	<b>57</b>
<b>Figura 27 – Cadastro de Colaboradores: interface para inclusão de novos usuários. Fonte: Autoria própria. . . . .</b>	<b>58</b>
<b>Figura 28 – Listagem de Colaboradores: visualização dos registros e gerenciamento dos usuários. Fonte: Autoria própria. . . . .</b>	<b>59</b>
<b>Figura 29 – Vinculação de Funcionários: interface para associação de colaboradores aos treinamentos. Fonte: Autoria própria. . . . .</b>	<b>60</b>

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
ERP	<i>Enterprise Resource Planning</i>
KPI	<i>Key Performance Indicator</i>
MVC	<i>Model-View-Controller</i>
MVP	<i>Minimum Viable Product</i>
NR	Norma Regulamentadora
PDF	Formato de Documento Portátil, do inglês <i>Portable Document Format</i>
RH	Recursos Humanos

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Objetivo Geral</b>	<b>11</b>
1.1.1	Objetivos específicos	11
<b>1.2</b>	<b>Justificativa</b>	<b>11</b>
<b>2</b>	<b>MATERIAIS E MÉTODOS</b>	<b>13</b>
<b>2.1</b>	<b>Materiais</b>	<b>13</b>
<b>2.2</b>	<b>Métodos</b>	<b>14</b>
2.2.1	Gerenciamento do processo	14
2.2.2	Controle de versão	15
2.2.3	Priorização de Requisitos	15
<b>3</b>	<b>ANÁLISE E PROJETO DO SISTEMA</b>	<b>17</b>
<b>3.1</b>	<b>Descrição Geral do Sistema</b>	<b>17</b>
<b>3.2</b>	<b>Levantamento de Requisitos</b>	<b>18</b>
3.2.1	Requisitos Não - Funcionais	18
3.2.2	Requisitos funcionais	19
<b>3.3</b>	<b>Histórias de Usuário e Protótipos de Tela</b>	<b>19</b>
<b>3.4</b>	<b>Modelagem do Banco de Dados</b>	<b>22</b>
<b>4</b>	<b>DESENVOLVIMENTO DO SISTEMA</b>	<b>25</b>
<b>4.1</b>	<b>Fluxo de Desenvolvimento</b>	<b>25</b>
<b>4.2</b>	<b>Escopo de Desenvolvimento</b>	<b>25</b>
4.2.1	Funcionalidades Implementadas	25
4.2.2	Funcionalidades Adicionais	26
<b>4.3</b>	<b>Configuração do Ambiente de Desenvolvimento</b>	<b>26</b>
4.3.1	Tecnologias e Versões	26
4.3.2	Principais Dependências (Gems)	26
4.3.3	Passos para Instalação	27
<b>4.4</b>	<b>Implementação de Funcionalidades</b>	<b>28</b>
4.4.1	Feature: Vinculação de Colaborador a Setor	28
4.4.2	Feature: Colaborador Altera o Status do Treinamento	30
<b>4.5</b>	<b>Fluxo de Atividades no Sistema</b>	<b>32</b>

4.5.1	Fluxo do Administrador . . . . .	32
4.5.2	Fluxo do Colaborador . . . . .	39
<b>4.6</b>	<b>Resultados Quantitativos . . . . .</b>	<b>41</b>
4.6.1	Métricas de Implementação . . . . .	41
4.6.2	Estrutura da Aplicação (Padrão MVC) . . . . .	42
4.6.3	Atendimento aos Requisitos . . . . .	42
<b>4.7</b>	<b>Resultados Qualitativos . . . . .</b>	<b>43</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>44</b>
5.1	<b>Trabalhos Futuros . . . . .</b>	<b>45</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>46</b>
	<b>APÊNDICE A HISTÓRIAS DE USUÁRIOS E PROTÓTIPOS DE TELA . . . . .</b>	<b>48</b>
	<b>A.1 Histórias de usuários: Geral . . . . .</b>	<b>48</b>
	<b>A.2 Histórias de usuários: : Usuário Administrador (<i>Must e Should</i>) . . . . .</b>	<b>49</b>
	<b>A.3 Histórias de usuários: Usuário Colaborador (<i>Must e Should</i>) . . . . .</b>	<b>53</b>
	<b>A.4 Protótipo de Telas: Área do Colaborador . . . . .</b>	<b>54</b>
	<b>A.5 Protótipo de Telas: Área do Administrador . . . . .</b>	<b>54</b>

## 1 INTRODUÇÃO

O investimento em treinamentos é essencial para garantir a segurança e a eficiência no ambiente de trabalho. Segundo a Associação Brasileira de Treinamento e Desenvolvimento, 94% das empresas brasileiras já investem em capacitação para seus colaboradores (AL1, 2024). Entretanto, apesar desse esforço, falhas na organização e no acompanhamento desses treinamentos ainda geram consequências graves, como acidentes de trabalho e baixa produtividade. Em alguns casos, embora os colaboradores estejam capacitados para realizar determinadas atividades, problemas relacionados ao controle inadequado dos treinamentos realizados pelos colaboradores geram efeitos negativos para as empresas.

Um caso real presenciado durante uma experiência de trabalho, ocorreu em uma empresa onde um colaborador sofreu um acidente que resultou na amputação de três dedos. Embora tivesse recebido treinamento, a ausência de um sistema eficiente de controle impossibilitou, inicialmente, a comprovação de sua capacitação. O processo de checagem foi lento e complicado devido ao armazenamento desestruturado de documentos físicos em um arquivo convencional. Além disso, a falta de controle sobre atualizações de treinamentos gerava inconsistências que comprometiam a segurança e as operações.

Essa situação reflete a necessidade de ferramentas adequadas para gerenciar as informações sobre os treinamentos realizados pelos colaboradores de uma empresa. Planilhas e documentos físicos são insuficientes para atender às demandas de um ambiente de trabalho moderno, onde a conformidade regulatória e a segurança dos processos são indispensáveis.

Neste sentido, o Training System foi idealizado para suprir essa lacuna. Trata-se de um sistema que organiza e automatiza a gestão de treinamentos realizados pelos colaboradores, permitindo às empresas: a) monitorar o histórico de capacitações, identificando rapidamente quem está apto para desempenhar atividades específicas; b) receber alertas automáticos sobre a necessidade de atualizações ou treinamentos; c) facilitar auditorias e consultas, com acesso rápido a informações organizadas digitalmente.

Além disso, o Training System foca nas necessidades das empresas, oferecendo ferramentas para garantir o cumprimento das normas de segurança e a conformidade com legislações, como a Norma Regulamentadora (NR)-12 (Norma Regulamentadora Nº 12 - Segurança no Trabalho em Máquinas e Equipamentos), que regula o trabalho com máquinas e equipamentos. Desta forma, além de auxiliar na redução dos riscos de acidentes, o sistema otimiza o processo de verificação dos treinamentos, melhora a organização documental e contribui para a tomada de decisões estratégicas. O resultado é uma gestão mais eficiente e segura, permitindo que as empresas se concentrem em seus objetivos principais sem comprometer a segurança no ambiente de trabalho. Assim, o Training System poderá proporcionar para as organizações um maior controle e eficiência na gestão de seus processos de capacitação.

## 1.1 Objetivo Geral

Desenvolver um sistema web para o controle dos treinamentos realizados por colaboradores de empresas, permitindo a identificação e acompanhamento de seu estado.

### 1.1.1 Objetivos específicos

- Levantar e analisar os requisitos necessários para o desenvolvimento do sistema, considerando as necessidades do controle de treinamentos.
- Projetar e desenvolver uma interface web que permita o acesso de administradores e colaboradores, com autenticação e funcionalidades adequadas para cada perfil.
- Implementar funcionalidades para cadastro, gerenciamento e vinculação de colaboradores e treinamentos, centralizando as informações em um único ambiente.
- Criar recursos para controle do status dos treinamentos, possibilitando o acompanhamento do histórico e o registro das ações realizadas no sistema.
- Desenvolver mecanismos para a auxiliar em visualizações analíticas, além da documentação do projeto, facilitando a compreensão e manutenção do sistema.

## 1.2 Justificativa

O desenvolvimento do Training System surgiu a partir de uma necessidade real observada no ambiente de trabalho após um acidente e da dificuldade de acesso e organização dos registros de treinamento dos colaboradores. A ausência de um sistema estruturado para armazenar e gerenciar essas informações pode resultar em atrasos na verificação dos treinamentos e impactar a tomada de decisões em situações que exigem agilidade. Além da organização dos registros, a proposta do sistema também considera a relevância da prevenção no ambiente corporativo. Um acompanhamento mais estruturado dos treinamentos pode contribuir para um controle mais eficiente da capacitação dos funcionários, auxiliando as empresas na adequação às normas de segurança e aos procedimentos internos. Inicialmente, a necessidade foi abordada por meio de planilhas para estruturar os dados e facilitar a gestão das informações. Agora, a evolução para um sistema web visa aprimorar essa abordagem, trazendo uma solução mais acessível e integrada. A Figura 1 exemplifica a tela de *Dashboard* da planilha, onde é possível visualizar a área de atuação do colaborador, consultar os treinamentos associados à ele e acompanhar o status atual de cada treinamento.

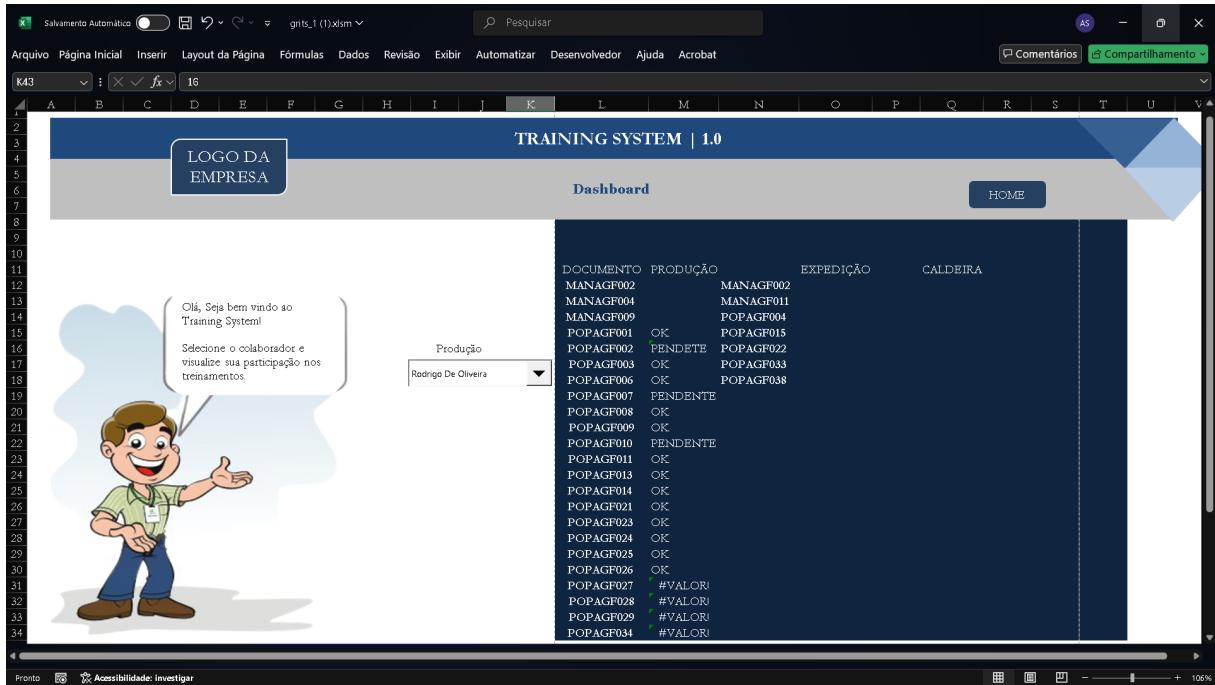


Figura 1 – Tela de dashboard do Training System com visualização por colaborador.

A partir da implementação dessa planilha, a empresa conseguiu maior agilidade na busca por informações em comparação com os documentos físicos. Contudo, o uso das planilhas também revelou novas limitações. Entre os principais desafios estavam: a) Falta de praticidade no gerenciamento completo dos dados; b) Impossibilidade de registrar acessos ou alterações feitas nos históricos; c) Dificuldade em associar diretamente os documentos aos treinamentos realizados e confirmar presenças. Esses problemas geraram insegurança durante as auditorias, pois não havia como garantir a rastreabilidade das alterações ou comprovar a validade das informações registradas.

A justificativa para o desenvolvimento de um sistema web de controle de treinamentos realizados por colaboradores de uma empresa está baseada na necessidade de otimizar e centralizar a gestão de capacitações e competências dos colaboradores. Esse tipo de sistema deve auxiliar na supervisão eficaz dos treinamentos, permitindo que a empresa mantenha um registro confiável das qualificações de seus colaboradores e identifique facilmente a falta de treinamentos e capacitações que precisam ser realizados para atender às demandas da empresa.

## 2 MATERIAIS E MÉTODOS

Este capítulo descreve as ferramentas e métodos aplicados no desenvolvimento do Training System, um sistema proposto para o controle de treinamentos realizados por colaboradores.

### 2.1 Materiais

Os materiais incluem ferramentas, tecnologias e ambientes de desenvolvimento que foram utilizados para uma maior eficiência, organização e qualidade do projeto. A seguir, estão listados e descritos os principais recursos que foram empregados:

- Ruby on Rails: O Framework Ruby on Rails foi escolhido como a principal tecnologia para o desenvolvimento do sistema web devido à sua agilidade e capacidade de simplificar a construção de aplicações robustas e escaláveis. Rails segue o padrão *Model-View-Controller* (MVC), que organiza o código em camadas, facilitando a manutenção e expansão do sistema (Ruby on Rails, 2024).
- SQLite: O banco de dados SQLite foi selecionado para o projeto, pois atende às necessidades de armazenamento de forma leve e prática. Ele é ideal para aplicações de pequeno a médio porte, possibilitando o gerenciamento eficiente de informações de colaboradores, treinamentos e registros históricos (SQLite, 2024).
- Docker: O uso do Docker permite a criação de ambientes isolados e padronizados, eliminando problemas de dependências entre sistemas. A aplicação foi conteinerizada com Docker, permitindo que funcione de maneira idêntica em diferentes ambientes de desenvolvimento e produção (Docker Inc., 2024).
- Figma: O Figma foi utilizado para a prototipação da interface do sistema. A ferramenta possibilitou a criação de protótipos interativos, que foram validados antes da implementação, oferecendo uma experiência de usuário eficiente e alinhada às expectativas (Figma, 2024).
- GitHub: O repositório do projeto foi hospedado no GitHub para controle de versão. A utilização do Git permitiu o rastreamento das alterações no código-fonte, facilitando o gerenciamento do desenvolvimento e mantendo a segurança do histórico do projeto (GitHub, 2024).
- ClickUp: O ClickUp foi empregado para a gestão de tarefas e cronograma do projeto, utilizando práticas da metodologia ágil Scrum. As tarefas foram organizadas em Sprints, com a definição de prazos e prioridades, permitindo um acompanhamento eficiente do progresso (ClickUp, 2024).

- MkDocs: Para a documentação do projeto, foi utilizada a ferramenta MkDocs, que serve para a criação de documentação estruturada e de fácil navegação. O MkDocs permite a geração de páginas estáticas a partir de arquivos em Markdown, possibilitando uma organização clara e acessível da documentação técnica do sistema (MkDocs, 2024).

## 2.2 Métodos

No planejamento inicial do desenvolvimento do Training System, estavam previstos o uso da metodologia ágil Scrum, da estratégia de versionamento GitFlow e do método de priorização MoSCoW. Contudo, no decorrer da execução, foi necessário adaptar algumas dessas práticas à realidade disponível de tempo e recursos, de forma que apenas parte do que havia sido planejado pôde ser efetivamente aplicada. Assim, o desenvolvimento combinou fundamentos de organização de tarefas e priorização estruturada com um processo de implementação linear e individual, mais compatível com as condições enfrentadas.

### 2.2.1 Gerenciamento do processo

A metodologia ágil Scrum é amplamente reconhecida por sua capacidade de estruturar o trabalho em ciclos iterativos (*Sprints*) e por se apoiar em três pilares fundamentais: transparência, inspeção e adaptação (SUTHERLAND, 2014; Atlassian, 2025). No planejamento inicial, o Scrum seria adotado como referência para conduzir o desenvolvimento do projeto. Na prática, entretanto, a estrutura completa do Scrum precisou ser adaptada. Devido à rotina de trabalho em paralelo ao projeto, que dificultou a marcação de reuniões frequentes com o orientador, não foi possível realizar os ciclos de *Sprints* com seus eventos formais, como a *Daily Scrum*.

Apesar disso, os princípios essenciais do Scrum foram mantidos. A lista de funcionalidades e tarefas do projeto foi organizada em um *Product Backlog*<sup>1</sup>, que foi priorizado continuamente. O pilar da transparência foi aplicado por meio de um quadro *Kanban*<sup>2</sup> na ferramenta ClickUp, que permitia visualizar o fluxo de trabalho e o status de cada entrega. Já os pilares de inspeção e adaptação foram aplicados na autogestão do desenvolvimento: cada tarefa era inspecionada individualmente ao ser concluída e, caso surgissem impedimentos técnicos, o plano de execução para as funcionalidades seguintes era reavaliado.

Dessa forma, o quadro Kanban não foi um substituto, mas a ferramenta escolhida para aplicar os conceitos ágeis do Scrum de maneira flexível e compatível com as limitações do projeto.

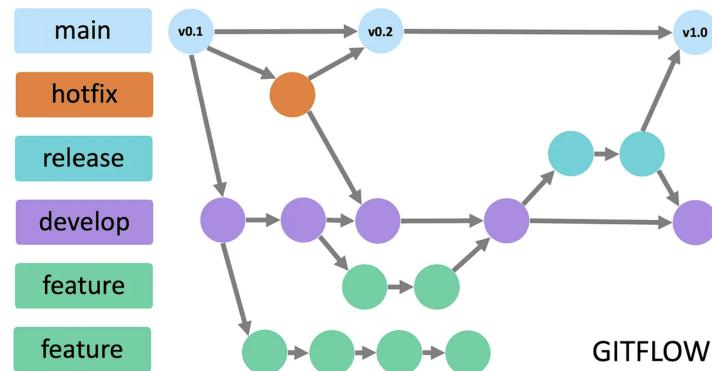
<sup>1</sup> No contexto do Scrum, o *Product Backlog* é uma lista única, priorizada e dinâmica de tudo o que é conhecido e necessário para o produto. Funciona como a única fonte de trabalho para a equipe.

<sup>2</sup> Kanban é um método de gerenciamento visual que utiliza um quadro (físico ou digital) para representar as etapas de um fluxo de trabalho. Seu objetivo é otimizar o fluxo de entrega, tornando o progresso e os gargalos visíveis.

### 2.2.2 Controle de versão

O modelo GitFlow organiza o versionamento em diferentes *branches* com papéis específicos, como *main*, *develop*, *feature* e *release*, facilitando a colaboração em equipe e o gerenciamento de versões em projetos de maior complexidade (DRIESSEN, Vincent, 2010). Inicialmente, essa abordagem seria utilizada no desenvolvimento do *Training System*.

Embora o planejamento inicial previsse o uso integral do GitFlow com múltiplas *branches* especializadas (*feature*, *develop*, *release*, *hotfix*), a realidade do desenvolvimento individual e as restrições de tempo conduziram a uma estratégia simplificada e mais pragmática. O repositório foi mantido apenas na *branch* principal (*main*), com *commits* pontuais de segurança (*checkpoints*) para prevenir a perda de código. Assim, não houve ramificações formais ou fluxo de integração contínua, mas sim uma estratégia simplificada que atendeu às necessidades do projeto. A Figura 2 apresenta o fluxo original do GitFlow, utilizado apenas como referência teórica.



**Figura 2 – Fluxo de trabalho do GitFlow. Fonte:** (PX, 2023).

### 2.2.3 Priorização de Requisitos

O método de priorização MoSCoW, criado por Dai Clegg na década de 1990, é utilizado para classificar requisitos de acordo com sua relevância para o projeto. Ele divide as funcionalidades em quatro categorias: *Must have* (deve ter), *Should have* (deveria ter), *Could have* (poderia ter) e *Won't have for now* (não terá por agora) (SEBRAE, 2022). A Figura 3 ilustra essa estrutura.

Diferente das demais metodologias previstas, o MoSCoW foi efetivamente adotado e teve papel fundamental no direcionamento do desenvolvimento. Sua aplicação permitiu identificar com clareza quais requisitos seriam obrigatórios, quais poderiam ser postergados e quais não seriam implementados no momento, considerando as limitações de tempo do projeto. Dessa forma, o método garantiu foco nas entregas essenciais e contribuiu para a organização prática das tarefas.



Figura 3 – Estrutura do método MoSCoW. Fonte: (SEBRAE, 2022).

Tabela 1 – Método MoSCoW para priorização de requisitos

Categoria	Tradução	Descrição
<b>M – Must have</b>	Deve ter	Requisitos essenciais e obrigatórios para o sistema. Sem eles, o produto não atende ao objetivo mínimo ou não pode ser entregue.
<b>S – Should have</b>	Deveria ter	Requisitos importantes, mas não críticos. Se não forem implementados, ainda é possível entregar o sistema, mas com menor valor.
<b>C – Could have</b>	Poderia ter	Requisitos desejáveis, mas não prioritários. Se houver tempo e recursos, podem ser incluídos; caso contrário, não comprometem o projeto.
<b>W – Won't have (for now)</b>	Não terá (por enquanto)	Requisitos que foram deliberadamente deixados de fora do escopo atual, mas podem ser considerados em versões futuras.

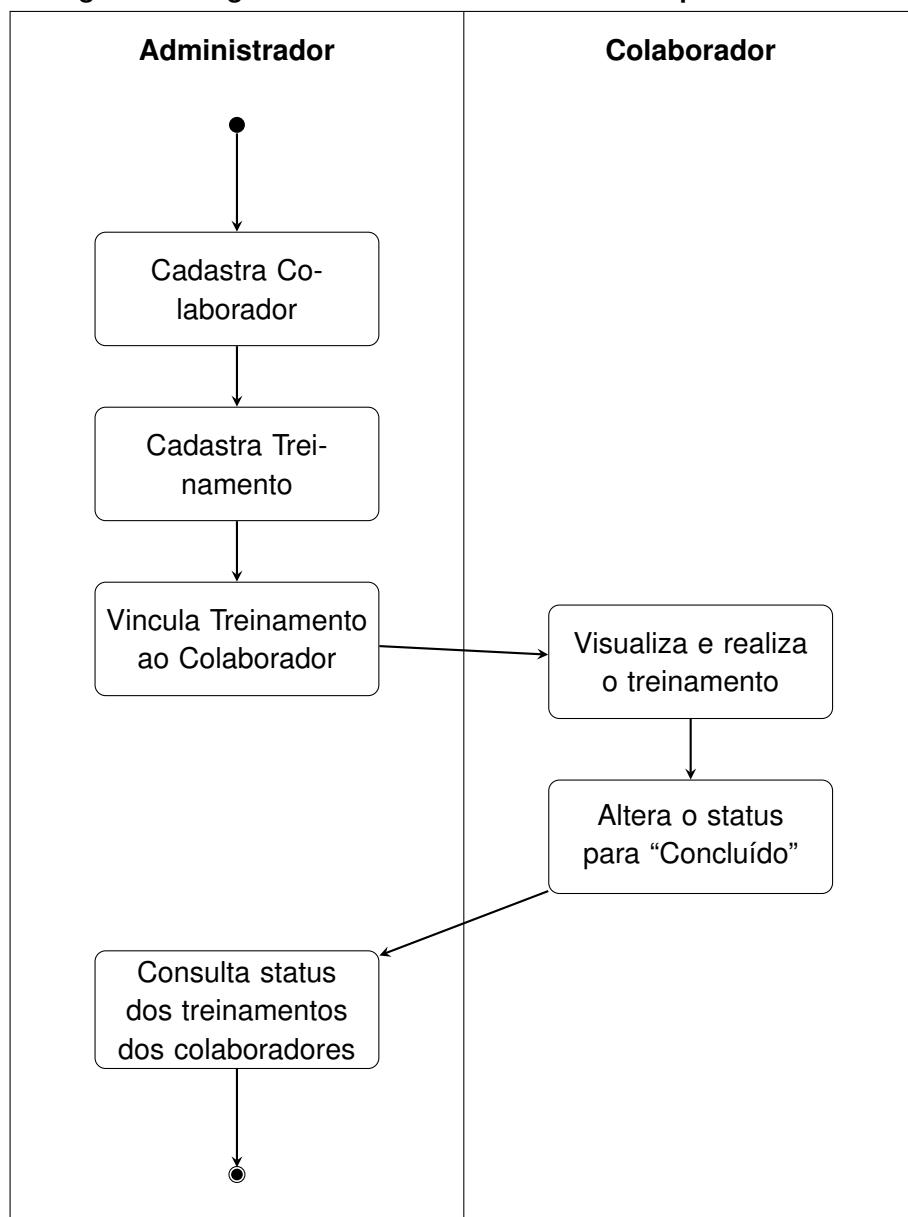
### 3 ANÁLISE E PROJETO DO SISTEMA

Este capítulo dedica-se à apresentação da análise e do projeto do sistema proposto. O objetivo é expor os artefatos e as decisões de projeto que fundamentam a solução desenvolvida.

#### 3.1 Descrição Geral do Sistema

O Training System foi desenvolvido para controlar os treinamentos realizados por colaboradores das empresas, oferecendo controles centralizados, conformidades legais, redução de erros manuais, assistência em auditorias e relatórios. O sistema trabalha seguindo um fluxo determinado, que se estabelece no momento do cadastro de colaboradores até o resultado das conclusões dos treinamentos. O diagrama de atividades da Figura 4 ilustra este fluxo.

**Figura 4 – Diagrama de Atividades do Fluxo Principal do Sistema.**



Além disso, convém apresentar as seguintes funcionalidades do sistema.

- Controle centralizado: é possível armazenar todas as informações sobre os treinamentos realizados pelos colaboradores, incluindo datas, documentos utilizados, quem participou e o instrutor responsável por conduzir o conteúdo. Permitindo assim, que quando houver a necessidade de alguma informação, seja possível acessar o histórico e encontrar o que necessita de modo completo.
- Conformidade com as normas: o sistema possui funções de alertas, o qual verifica o prazo de validade dos documentos de treinamentos e dispara alertas quando esses estiverem próximos das datas de vencimento, permitindo que assim não possua documentos com datas vencidas e cumpra com as regulamentações definidas pelo Ministério do Trabalho e Emprego.
- Relatórios: recurso de relatórios onde é possível a gestão obter informações rápidas, como identificar áreas ou equipes que necessitam de mais treinamentos, auxiliando ainda nas tomadas de decisões.
- Auditorias: o sistema pode ser utilizado como uma fonte para revisões internas pelos administradores. Além de ser um método para comprovação dos treinamentos, e conformidade com o fluxo de organização da empresa.

## 3.2 Levantamento de Requisitos

Este capítulo apresenta os principais requisitos do desenvolvimento do sistema de gerenciamento de treinamentos. O objetivo é proporcionar uma compreensão clara das suas funcionalidades e adequação aos requisitos levantados.

### 3.2.1 Requisitos Não - Funcionais

Ainda nas funcionalidades e benefícios do sistema, é possível destacar que ele pode contribuir para a redução de erros manuais, pois, quando uma organização é feita em planilhas, documentos físicos ou em softwares não apropriados, podem ocorrer omissões de datas importantes. Por exemplo, se alguém não autorizado apagar informações ou ainda fazer alterações em dados relevantes, não é possível averiguar quem foi o responsável. Além disso, podem ocorrer perdas de informações e, o principal, a falta de organização. Com o sistema, esses erros podem ser evitados evitados, assegurando que os dados fossem registrados corretamente e que os prazos de treinamento e os requisitos para desempenhar cada função na empresa fossem atendidos.

A Tabela 2 apresenta um resumo desses requisitos

**Tabela 2 – Requisitos Não Funcionais e seus Atributos de Qualidade.**  
**Fonte: Autoria própria.**

Requisito	Atributo de Qualidade
Redução de erros manuais	Confiabilidade
Controle de acesso (quem pode alterar/apagar informações)	Segurança
Evitar perdas de informações	Integridade/Armazenamento Confiável
Assegurar que os prazos sejam cumpridos	Confiabilidade e Disponibilidade de Informações

### 3.2.2 Requisitos funcionais

Nesta seção, são detalhados os principais requisitos funcionais do sistema. Cada requisito foi classificado utilizando o método para priorização de requisitos MoSCoW. As Tabelas 3, 4 e 5 apresentam, respectivamente, os requisitos gerais (Usuário Administrador e Colaborador), os requisitos destinados especificamente ao Usuário Colaborador e aqueles destinados ao Usuário Administrador. Esse levantamento estrutura as principais funcionalidades para cada perfil de usuário, classificadas pela técnica *MoSCoW* (*Must*, *Should*, *Could* e *Won't*).

**Tabela 3 – Requisitos gerais do sistema (Usuário Administrador e Colaborador).**  
**Fonte: Autoria própria.**

Requisito	Descrição detalhada	MoSCoW
Tela Inicial	Exibir uma tela inicial contendo tutoriais e manuais de uso para facilitar a navegação dos usuários. Essa tela será acessível para todos os usuários do sistema.	Must
Autenticação de usuário	Validar o tipo de usuário (colaborador ou administrador) e direcionar o acesso às funcionalidades apropriadas.	Must
Guarda de rotas	Restringir o acesso a áreas específicas do sistema, garantindo que apenas usuários autenticados possam acessá-las.	Must
Notificações e alertas	Enviar lembretes automáticos sobre treinamentos pendentes, prazos de vencimento e novas inscrições.	Should
Integração com outros sistemas	Permitir integração com sistemas de Recursos Humanos (RH) e <i>Enterprise Resource Plannings</i> (ERPs) para sincronização de dados.	Won't
Mobile-friendly	Garantir que o sistema seja responsivo e funcione corretamente em dispositivos móveis.	Won't
Assinatura digital	Implementar um sistema de assinatura digital para validar registros de conclusão de treinamento.	Won't
Reconhecimento facial	Validar a presença dos colaboradores nos treinamentos por meio de reconhecimento facial.	Won't
Suporte multilíngue	Disponibilizar o sistema em múltiplos idiomas para maior acessibilidade.	Won't
Filtro avançado de treinamentos	Permitir busca detalhada por treinamentos aplicando filtros como colaborador, data, instrutor e status.	Won't

### 3.3 Histórias de Usuário e Protótipos de Tela

A partir dos requisitos levantados, foram elaboradas histórias de usuário com base nos itens *Must* e *Should* de cada perfil (administrador e/ou colaborador), de acordo com a prioriza-

**Tabela 4 – Requisitos para colaboradores (Usuário Colaborador).****Fonte: Autoria própria.**

Requisito	Descrição detalhada	MoSCoW
Listagem de treinamentos para colaboradores	Permitir que os colaboradores visualizem todos os treinamentos disponíveis, destacando o status (pendente, em andamento, concluído).	Must
Detalhes do treinamento	Exibir informações completas de um treinamento, incluindo descrição, datas, instrutor responsável e status. Disponível para todos os usuários.	Must
Perfil do colaborador	Permitir que o colaborador visualize e edite suas informações básicas, como nome, setor e cargo.	Must
Vínculo de colaborador e treinamento	Associar treinamentos a colaboradores e registrar os documentos obrigatórios utilizados no processo.	Must
Dashboard do colaborador	Exibir um painel com treinamentos pendentes, concluídos e próximos vencimentos, incluindo gráficos de progresso.	Won't
Certificação automática	Gerar certificados automaticamente para colaboradores que concluírem um treinamento.	Won't

ção MoSCoW. Cada história descreve o papel do usuário, a funcionalidade desejada e o valor que ela agrega, além de critérios de aceitação no formato Dado – Quando – Então.

Essas histórias contemplam funcionalidades básicas compartilhadas pelos perfis de administrador e colaborador. O objetivo é garantir que ambos possam acessar a tela inicial com tutoriais, efetuar login de forma segura, ter acesso restrito a funcionalidades de acordo com seu nível de permissão e receber notificações pertinentes a treinamentos e prazos.

A seguir, é apresentado um exemplo de história de usuário que atende ao visualizações analíticas. As demais histórias estão descritas no Apêndice A.

#### Feature: Listagem de colaboradores

**Como** administrador, **quero** visualizar todos os colaboradores cadastrados **para** aplicar filtros e acompanhar o status de cada um.

**Dado** que sou um administrador autenticado,

**Quando** acesso a lista de colaboradores,

**Então** devo ver todos os colaboradores com a opção de filtrar e ordenar por nome, setor ou status.

Para capturar os detalhes dos dados de cada funcionalidade do sistema, foram criados protótipos de tela. A seguir, é apresentado um exemplo de tela que corresponde a história de usuário “Listagem de Colaboradores” (seção A.2). As demais telas estão apresentadas no Apêndice A.

**Tabela 5 – Requisitos para administradores (Usuário Administrador).****Fonte: Autoria própria.**

Requisito	Descrição detalhada	MoSCoW
Registros de histórico	Registrar todas as ações realizadas no sistema, incluindo alterações de status e edições de cadastros, para fins de auditoria. Apenas administradores poderão visualizar esse histórico.	Must
Alteração de status do treinamento	Permitir que administradores modifiquem o status de um treinamento (pendente, em andamento, concluído).	Must
Cadastro de colaboradores	Permitir que administradores registrem novos colaboradores no sistema.	Must
Cadastro de treinamentos	Permitir que administradores registrem novos treinamentos, definindo datas, instrutores e critérios de participação.	Must
Listagem de colaboradores	Permitir que administradores visualizem a lista de colaboradores cadastrados, aplicando filtros e ordenação por nome, setor e status.	Must
Listagem de treinamentos para administradores	Permitir que administradores consultem todos os treinamentos cadastrados, com filtros por nome, data e instrutor.	Must
Editar colaboradores	Permitir que administradores atualizem os dados cadastrais dos colaboradores, como nome, cargo e setor.	Must
Editar treinamentos	Permitir que administradores editem as informações de um treinamento, incluindo datas e participantes.	Must
Excluir treinamentos	Permitir que administradores removam treinamentos, garantindo que não haja exclusão indevida de dados críticos.	Must
Excluir colaboradores	Permitir que administradores removam colaboradores inativos, preservando o histórico de treinamentos realizados.	Must
Controle de acesso	Definir perfis de acesso para administradores e colaboradores, garantindo que cada usuário tenha permissões adequadas.	Must
Perfil do administrador	Permitir que administradores visualizem e editem suas informações e gerenciem contas de outros usuários.	Must
Controle de versão dos treinamentos	Registrar e permitir a atualização das versões dos treinamentos oferecidos, garantindo que colaboradores sempre tenham acesso às versões mais recentes.	Should
Padronização de documentos obrigatórios	Automatizar a atribuição de treinamentos obrigatórios para novos colaboradores.	Should
Supporte a auditorias	Permitir que administradores exportem dados para revisões internas e conformidade com normas regulatórias.	Could
Exportação de relatórios	Gerar relatórios em formatos como PDF e Excel para facilitar a análise e compartilhamento.	Could
Geração de relatórios	Criar relatórios detalhados sobre treinamentos realizados e pendentes, com filtros avançados.	Won't
Dashboards interativos	Exibir indicadores de desempenho, como taxa de conclusão e pendências dos treinamentos.	Won't

The screenshot shows the 'Visão Geral de Treinamentos' (General Training View) page. On the left, a sidebar menu includes 'Home', 'Meus Treinamentos', 'Meu Perfil', 'ADMINISTRAÇÃO' (with 'Atribuir Treinamentos' and 'Visão Geral Treinamentos' selected), 'Setores Cadastrados', 'Treinamentos Cadastrados', and 'Usuários'. The main content area has a title 'Visão Geral de Treinamentos' and a breadcrumb 'Treinamentos > Visão Geral > Home'. It features four summary boxes: 'TOTAL ATRIBUIÇÕES 13', 'USUÁRIOS COM TREINAMENTOS 7', 'OBRIGATÓRIOS 2', and 'CONCLUÍDOS 4'. Below these are filtering options for 'Colaborador', 'Treinamento', 'Status', and 'Obrigatório', with buttons for 'Filtrar' and 'Limpar'. A table lists training sessions for users: Administrador (Segurança no trabalho, Pendente, Detalhes), André Barboza (Tecnologia (+1), Treinamento em ruby, Em Andamento, Detalhes), and André Barboza (Tecnologia (+1), Treinamento teste, Pendente, Detalhes).

**Figura 5 – Listagem dos Treinamentos Colaboradores: visualização dos registros de treinamentos dos usuários. Fonte: Autoria própria.**

### 3.4 Modelagem do Banco de Dados

A modelagem do banco de dados foi um passo fundamental para estruturar as informações do Training System. A abordagem utilizada pelo Ruby on Rails, através do Active Record, permite uma representação clara das entidades do sistema e seus relacionamentos. A Figura 6 apresenta o modelo relacional do sistema, que ilustra as principais entidades, seus relacionamentos e como os dados são organizados.

As tabelas centrais do domínio da aplicação são:

- **users**: armazena os dados de todos os usuários, diferenciando administradores e colaboradores através do campo booleano `is_admin`. controla informações de autenticação, confirmação de conta e status (ativo/inativo).
- **departments** : Representa os setores da empresa, contendo nome e descrição.
- **trainings** : define a entidade principal de um treinamento, com título, descrição e duração.
- **training\_versions** : uma das principais decisões de projeto, esta tabela permite que cada treinamento possua múltiplas versões, controlando o número da versão, data de lançamento, status (ativo/inativo) e conteúdo. Um treinamento (`training`) possui muitas versões (`training_versions`).

Os relacionamentos entre essas entidades são gerenciados por tabelas de associação (*join tables*), permitindo a integridade e a flexibilidade do modelo:



Figura 6 – Modelo Relacional do Banco de Dados. Fonte: Gerado com dbdiagram.io.

- `user_departments`: associa um usuário a um setor, implementando um relacionamento N-para-N (muitos-para-muitos).
- `department_trainings`: Vincula um treinamento a um setor, também em um relacionamento N-para-N, e inclui um campo `mandatory` para definir a obrigatoriedade do treinamento para aquele setor.
- `user_training_versions`: É a tabela mais detalhada, pois representa a matrícula de um usuário em uma versão específica de um treinamento. Ela registra o status individual do treinamento para o usuário (pendente, concluído, etc.), datas importantes e se o treinamento foi atribuído como obrigatório.

## 4 DESENVOLVIMENTO DO SISTEMA

Este capítulo detalha o processo de construção do Training System, desde a organização das tarefas até a implementação efetiva das funcionalidades. O objetivo é apresentar as ferramentas, as estratégias e os desafios encontrados durante a transformação da análise de projeto em um sistema web funcional.

### 4.1 Fluxo de Desenvolvimento

O desenvolvimento do sistema foi guiado por uma abordagem adaptada aos princípios ágeis. O fluxo de trabalho para cada funcionalidade seguia um ciclo consistente: uma tarefa era selecionada no ClickUp e movida para o estado “Em Andamento”. Em seguida, o desenvolvimento era realizado de forma integrada, construindo a lógica do *backend* em Ruby on Rails e, na sequência, a interface no *frontend*. Após a implementação, cada funcionalidade era testada manualmente para garantir seu correto funcionamento. Ao ser concluída e validada, a tarefa correspondente no ClickUp era movida para o estado “Concluído”, garantindo a rastreabilidade e a organização do progresso do projeto.

### 4.2 Escopo de Desenvolvimento

Utilizando o método MoSCoW para a priorização de requisitos, todas as funcionalidades classificadas como *Must* e *Should* foram cadastradas como tarefas na ferramenta de gerenciamento de projetos ClickUp. O escopo do projeto foi definido com base na priorização MoSCoW (detalhada nas Tabelas 3, 4 e 5). O foco foi garantir a entrega de um (*Minimum Viable Product* (MVP) - Minimum Viable Product) que atendesse a todas as necessidades essenciais do sistema.

#### 4.2.1 Funcionalidades Implementadas

Foram implementadas todas as funcionalidades classificadas como *Must have*, visando a operação central do sistema. Isso inclui:

- **Funcionalidades gerais:** Tela inicial, autenticação de usuário e controle de acesso por rotas.
- **Funcionalidades do colaborador:** Listagem e detalhamento de treinamentos, gestão do próprio perfil e vínculo automático com treinamentos obrigatórios.
- **Funcionalidades de administrador:** *Create, Read, Update, Delete* (CRUD) completo de colaboradores (listagem, edição, exclusão), treinamentos (listagem, edição, exclu-

são) e setores, além da capacidade de vincular treinamentos a setores, editar o status de treinamentos dos colaboradores e visualizar o histórico de ações.

Também foram implementadas todas as funcionalidades *Should have*, como o controle de versão de treinamentos. A funcionalidade de “Notificações e Alertas” foi parcialmente atendida: o sistema exibe alertas visuais na interface para o usuário sobre treinamentos pendentes, embora o envio ativo de notificações por e-mail não tenha sido incluído nesta versão.

#### 4.2.2 Funcionalidades Adicionais

Durante o desenvolvimento, foi implementada uma funcionalidade não prevista no escopo inicial, mas que agrega valor à segurança do sistema: um fluxo de redefinição de senha por e-mail. Embora em formato de MVP, a funcionalidade permite que o usuário solicite uma nova senha caso esqueça a sua, enviada em um e-mail de recuperação.

### 4.3 Configuração do Ambiente de Desenvolvimento

Nesta seção, são detalhados os componentes técnicos e as ferramentas que compuseram o ambiente de desenvolvimento do Training System. São apresentadas as versões do software utilizado, as principais dependências do projeto e os passos para a configuração de um ambiente local para a execução da aplicação.

#### 4.3.1 Tecnologias e Versões

O projeto foi construído utilizando as seguintes versões de software:

- **Ruby:** versão 3.4.3
- **Rails:** versão 8.0.2
- **Banco de Dados:** SQLite 3

O código-fonte do projeto foi versionado com Git e hospedado na plataforma GitHub, disponível para consulta no seguinte endereço: <https://github.com/andrebsilva10/training-system.git>.

#### 4.3.2 Principais Dependências (Gems)

O gerenciamento de dependências do projeto foi realizado através do Bundler. O arquivo `Gemfile` define as bibliotecas utilizadas, das quais se destacam:

- **Puma:** Servidor web utilizado para executar a aplicação em ambiente de desenvolvimento.
- **Propshaft e TailwindCSS-Rails:** Compõem o sistema de gestão de assets, com o TailwindCSS sendo o framework do *Cascading Style Sheets* (CSS) utilizado para a estilização das interfaces.
- **Hotwire (Turbo e Stimulus):** Conjunto de bibliotecas para o desenvolvimento de interfaces interativas, permitindo atualizações parciais da página sem a necessidade de recarregamentos completos.
- **Bcrypt:** Biblioteca utilizada para a segurança das senhas dos usuários, implementando sua criptografia antes do armazenamento no banco de dados.
- **Image Processing:** Dependência utilizada pelo Active Storage para o processamento e manipulação de imagens, como no caso dos avatares de perfil dos usuários.

#### 4.3.3 Passos para Instalação

Para configurar e executar o projeto em um ambiente de desenvolvimento local, os seguintes passos podem ser seguidos a partir do terminal:

1. Clonar o repositório do projeto:

```
git clone
```

```
https://github.com/andrebsilva10/training-system.git
```

2. Navegar para o diretório do projeto:

```
cd training-system
```

3. Instalar as dependências (gems) definidas no `Gemfile`:

```
bundle install
```

4. Criar a estrutura do banco de dados local:

```
rails db:create
```

5. Executar as migrações para criar as tabelas do banco de dados:

```
rails db:migrate
```

6. Iniciar o servidor web:

```
rails server
```

Após a execução desses comandos, a aplicação se torna acessível no navegador, geralmente pelo endereço <http://localhost:3000>.

#### 4.4 Implementação de Funcionalidades

Nesta seção, são detalhadas as implementações de um conjunto representativo de funcionalidades do Training System. O objetivo é demonstrar, por meio de exemplos práticos, como os requisitos e as histórias de usuário foram traduzidos em código e como as diferentes partes do sistema (visão, controle e modelo) interagem para entregar o valor esperado.

##### 4.4.1 Feature: Vinculação de Colaborador a Setor

Uma das funcionalidades centrais para a automação de processos no sistema é a capacidade de um administrador vincular um colaborador a um ou mais setores. Esta ação atende à história de usuário *“Como administrador, quero associar treinamentos a colaboradores”*, sendo um requisito *Must have*, e é crucial por automatizar a atribuição de treinamentos obrigatórios.

A implementação desta feature foi distribuída entre o **Controller**, que lida com a requisição do usuário, e os **Models**, que contêm a lógica de negócio principal. A seguir são apresentados os detalhes da implementação desta funcionalidade no padrão MVC.

- Interface e Controle (View & Controller)

A interface para esta ação está localizada na página de detalhes de cada usuário. O administrador visualiza os setores aos quais o colaborador já pertence e tem a opção de associá-lo a novos setores. Ao submeter o formulário de adição, uma requisição **POST** é enviada para a rota `/users/:id/add_department`, que é gerenciada pela **action add\_department** no **UsersController**.

O **Controller** executa os seguintes passos:

1. **Autorização:** Primeiramente, o filtro `before_action :authorize_admin` garante que apenas um usuário com perfil de administrador possa executar esta ação.

2. **Validação:** O sistema verifica se o colaborador já está associado ao setor selecionado para evitar duplicidade de registros.
3. **Criação da Associação:** Caso a validação seja bem-sucedida, uma nova entrada é criada na tabela de associação `user_departments`, efetivando o vínculo entre o usuário e o setor. O código responsável é:

```
@user.user_departments.create(department: department)
```

- Lógica de Negócio e Automação (Model)

A principal complexidade desta feature reside na automação da atribuição de treinamentos. Embora o `Controller` apenas crie a associação, a lógica de negócios foi desacoplada e implementada nos `models` para ser acionada em um momento oportuno (por exemplo, através de um `callback` ou um serviço).

O coração desta lógica está no `Model Department`, dentro do método `assign_trainings_to_user(user)`. Este método é projetado para executar a seguinte rotina:

1. Ele busca todos os treinamentos que estão marcados como obrigatórios (`mandatory: true`) para aquele setor específico.
2. Para cada treinamento obrigatório encontrado, o método localiza a sua versão ativa mais recente.
3. Por fim, ele cria um novo registro na tabela `user_training_versions`, associando a versão mais recente do treinamento ao colaborador, com o status inicial de `pending` (pendente).

Esta abordagem permite que, ao ser movido para um novo setor, um colaborador receba automaticamente todas as capacitações necessárias, eliminando a necessidade de intervenção manual do administrador e reduzindo a possibilidade de falhas.

Esta estrutura separa as responsabilidades, deixando o `Controller` encarregado do fluxo de requisições e o `Model` responsável pelas regras de negócio e pela consistência dos dados.

O principal desafio técnico foi modelar e implementar a lógica de automação de forma eficiente e desacoplada. A decisão de colocar a responsabilidade da atribuição de treinamentos no `Model Department` permitiu que a regra de negócio ficasse centralizada e reutilizável, separada das responsabilidades do `Controller`.

O resultado é uma funcionalidade robusta que simplifica um processo administrativo crítico. A automação da atribuição de treinamentos não apenas economiza tempo, mas também aumenta a conformidade da empresa com suas políticas de capacitação. A interface final para o gerenciamento de setores de um colaborador é apresentada na Figura 7.

The screenshot shows the 'Training system' application interface. At the top, there is a navigation bar with a user profile icon and the name 'André Barboza'. On the left, a sidebar lists various menu items: 'Home' (selected), 'Meus Treinamentos', 'Meu Perfil', 'Administração', 'Atribuir Treinamentos', 'Visão Geral Treinamentos', 'Setores Cadastrados', 'Treinamentos Cadastrados', and 'Usuários'. The main content area is titled 'Detalhes do Usuário - André Silva' and shows the user's status as 'Ativo'. Below this, there are tabs for 'Informações', 'Treinamentos', and 'Setores' (selected). The 'Setores' section lists two departments: 'Manutenção' (2 users) and 'Produção' (2 users), each with a 'Remover' (Remove) button. A 'Voltar para Usuários' (Back to Users) button is located in the top right corner of the main content area.

**Figura 7 – Tela de gerenciamento de setores de um colaborador.**

#### 4.4.2 Feature: Colaborador Altera o Status do Treinamento

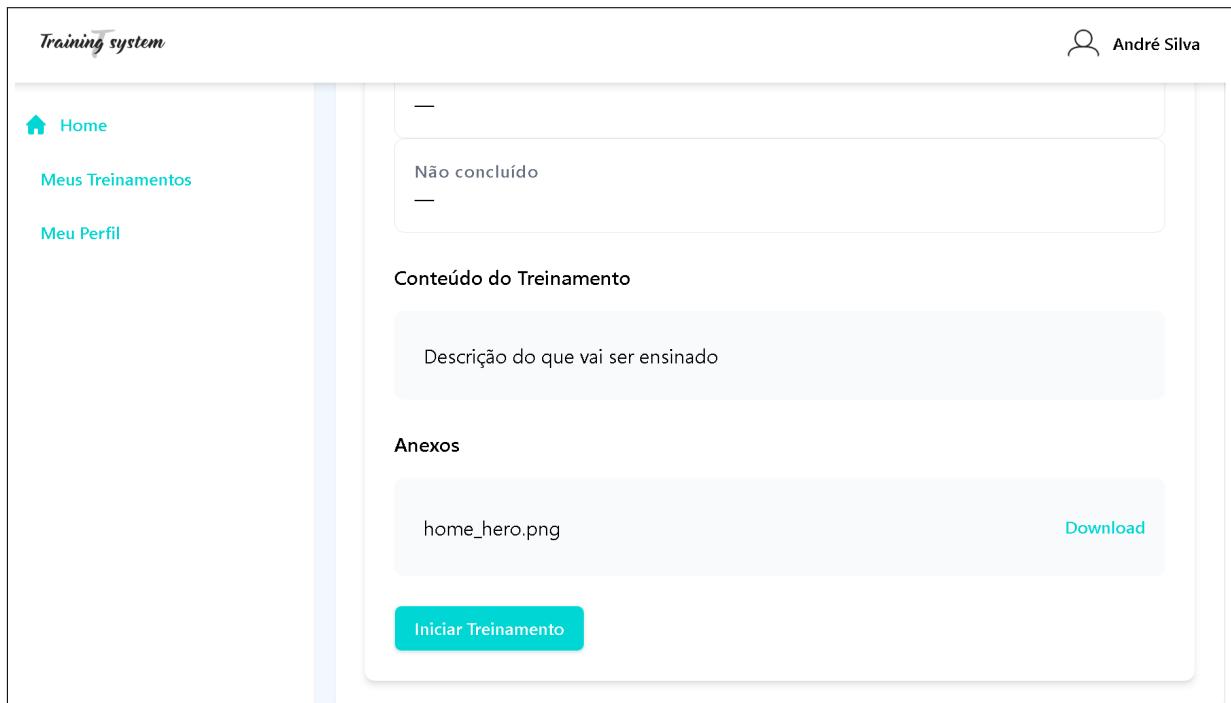
Esta funcionalidade é central para a experiência do colaborador, pois permite que ele gerencie ativamente o progresso de suas capacitações. Ela atende diretamente ao requisito funcional “Listagem de treinamentos para colaboradores” (Tabela 4) e à história de usuário “*Como colaborador, quero visualizar todos os treinamentos disponíveis para acompanhar meu progresso e identificar quais ainda estão pendentes, em andamento ou concluídos*” (Apêndice A), um requisito *Must have*.

A implementação dessa funcionalidade foi estruturada de modo que um treinamento progride por meio de uma sequência lógica de status: de `pending` (pendente) para `in_progress` (em andamento), e de `in_progress` para `completed` (concluído). Adicionalmente, um treinamento concluído é movido automaticamente para o status `expired` (expirado) assim que sua data de validade é ultrapassada. A seguir são apresentados os detalhes da implementação desta funcionalidade no padrão MVC.

- Interface e Controle (View & Controller)

O ponto de interação do colaborador é a tela de detalhes do treinamento (`show.html.erb`). A interface apresenta os botões de ação de forma condicional, guiando o usuário pelo fluxo correto.

- Se o treinamento está com o status `pending`, apenas o botão “**Iniciar Treinamento**” é exibido, conforme Figura 8 a seguir.



**Figura 8 – Tela de detalhes do treinamento pendente para o colaborador.**

- Se o treinamento está com o `in_progress`, o botão “Iniciar” é ocultado e o botão “Concluir Treinamento” é disponibilizado, conforme Figura 9.

Cada botão submete uma requisição `POST` para uma action específica no `UserTrainingVersionsController`. A action `start` é responsável por iniciar um treinamento, enquanto a action `complete` o finaliza.

A segurança é tratada pelo filtro `before_action :authorize_access`, que verifica se o usuário logado é o proprietário do registro de treinamento ou um administrador, previnindo que um usuário modifique o status de treinamentos de outros colaboradores.

- Lógica de Negócio (Model)

A lógica de estados e as regras de negócio associadas foram centralizadas no Model `UserTrainingVersion`.

- **Máquina de Estados:** o campo `status` foi implementado como um enum do Rails, que mapeia os nomes dos status (`pending`, `in_progress`, etc.)

para valores inteiros no banco de dados, o que otimiza o armazenamento e facilita as consultas.

- **Transição de Estados:** as actions no Controller validam o estado atual antes de permitir uma transição. Por exemplo, a action `complete` só executa a atualização se o treinamento estiver `in_progress?`, evitando que um treinamento pendente seja marcado como concluído diretamente.
- **Automação de Datas:** ao marcar um treinamento como concluído, o sistema executa duas automações importantes:
  1. Grava o timestamp atual no campo `completed_at`.
  2. Utiliza um callback `after_save` para acionar o método `update_due_date`. Este método calcula a data de expiração do treinamento, somando o período de validade (definido na versão do treinamento) à data de conclusão.

Esta estrutura separa as responsabilidades, deixando o Controller encarregado do fluxo de requisições e o Model responsável pelas regras de negócio e pela consistência dos dados. Um desafio na implementação foi projetar a máquina de estados de forma que as transições fossem seguras e que a interface do usuário refletisse com precisão o estado atual, mostrando apenas as ações permitidas para o colaborador em cada etapa do processo.

O resultado é uma funcionalidade que permite ao colaborador ter autonomia sobre seu progresso de capacitação. O fluxo guiado pela interface e validado no *backend* resulta em um processo de atualização de status claro e com baixa margem para erros de uso, conforme demonstrado na Figura 9.

## 4.5 Fluxo de Atividades no Sistema

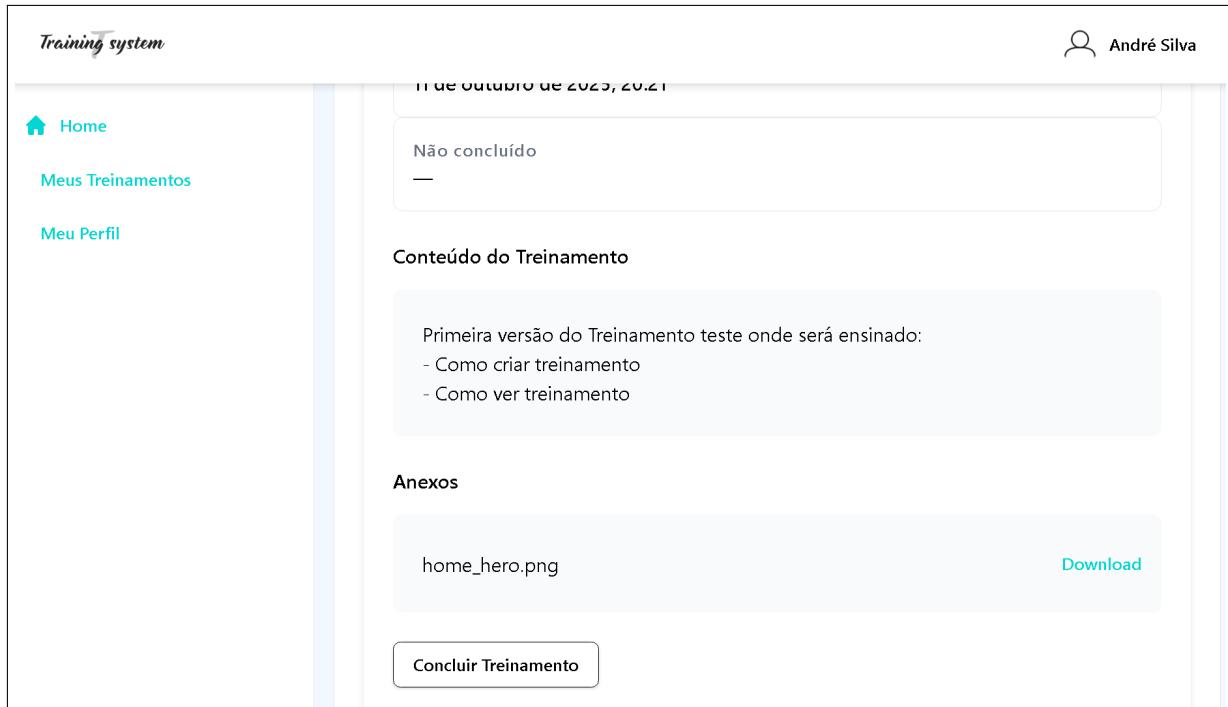
A utilização do sistema é dividida de acordo com os perfis de usuário: Administrador e Colaborador. O usuário acessa a tela de login e se autentica com suas credenciais de administrador ou colaborador, de acordo com a ilustração da Figura 10.

A seguir, é descrito o fluxo principal de atividades para cada um dos perfis, ilustrado com as telas finais da aplicação.

### 4.5.1 Fluxo do Administrador

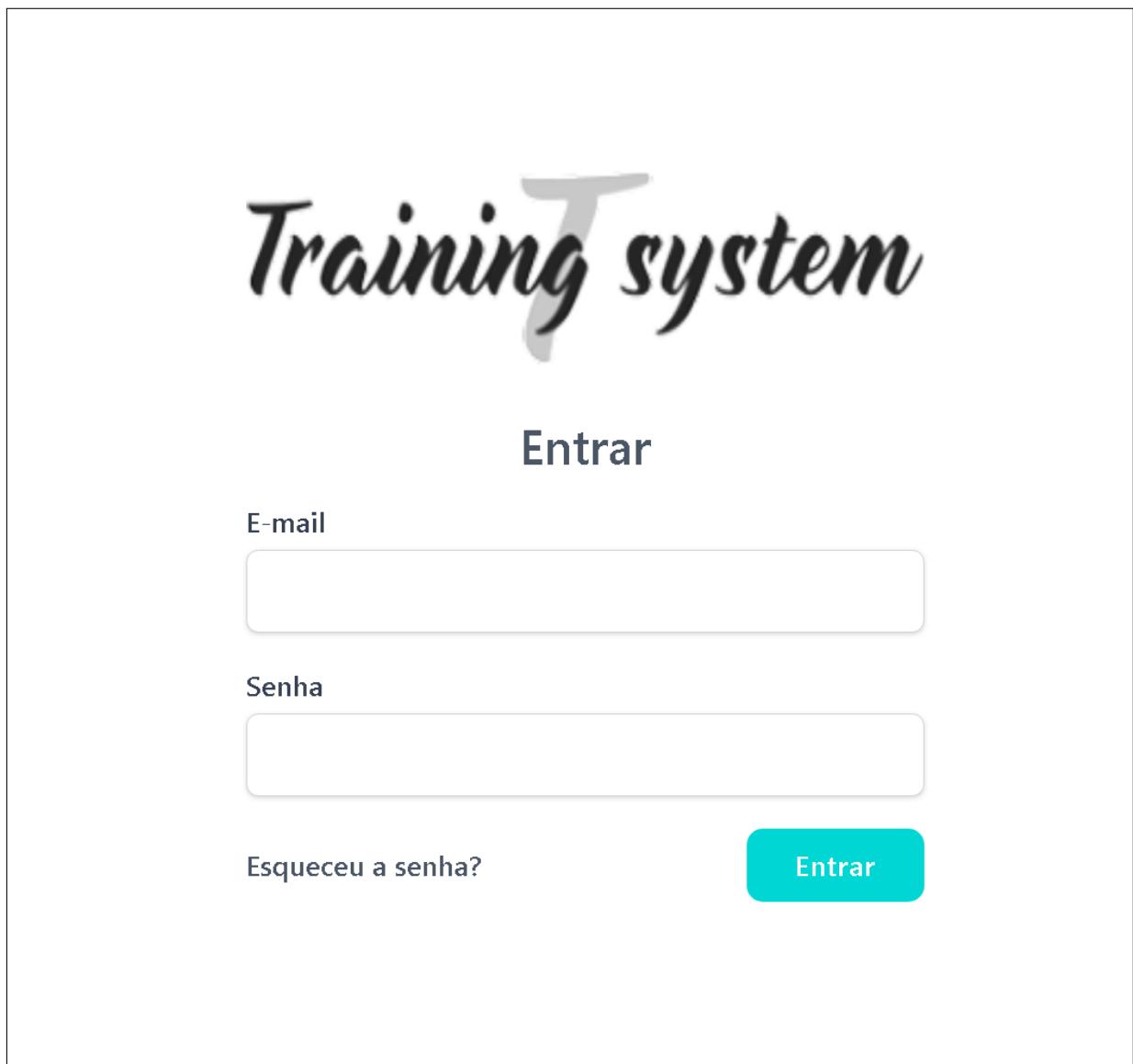
O administrador possui controle total sobre os dados do sistema, sendo responsável por configurar o ambiente, cadastrar usuários e treinamentos, e monitorar o progresso geral. O fluxo de atividades típico para este perfil, ilustrado na Figura 11, envolve a gestão completa do ciclo de vida dos treinamentos e colaboradores.

O fluxo detalhado, ilustrado com as telas da aplicação, é o seguinte:



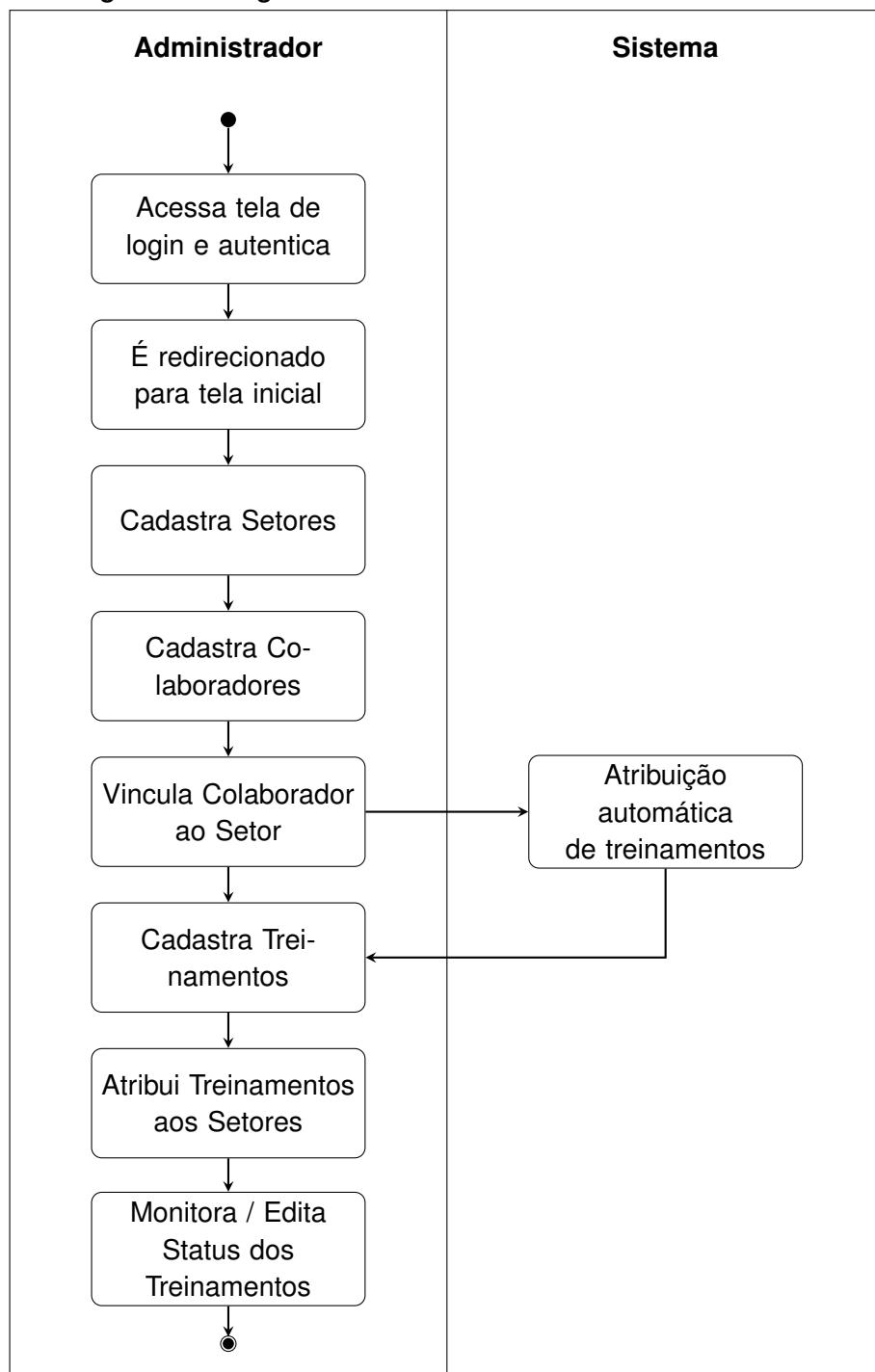
**Figura 9 – Tela de detalhes do treinamento em andamento para o colaborador.**

1. Após o login, o administrador é redirecionado para a tela inicial, que serve como painel de navegação, apresentada na Figura 12.
2. O administrador pode gerenciar os setores da empresa através de um CRUD completo. O primeiro passo é cadastrar os setores existentes, como ilustra a Figura 13.
3. Com os setores definidos, ele pode cadastrar novos colaboradores no sistema, preenchendo seus dados e definindo seu acesso, conforme exibido na Figura 14.
4. O administrador pode, então, vincular um colaborador a um setor. Neste momento, o sistema automaticamente atribui ao colaborador todos os treinamentos definidos como obrigatórios para aquele setor. A interface para esta ação é exibida na Figura 15.
5. Adicionalmente, o administrador gerencia os treinamentos, podendo cadastrar novos, editar existentes e atribuí-los a setores, definindo sua obrigatoriedade, conforme a tela da Figura 16.
6. Por fim, ele pode monitorar o status dos treinamentos de todos os colaboradores e, se necessário, editar o status manualmente, conforme ilustrado na Figura 17.



**Figura 10 – Tela de login.**

Figura 11 – Diagrama de Atividades - Fluxo do Administrador.



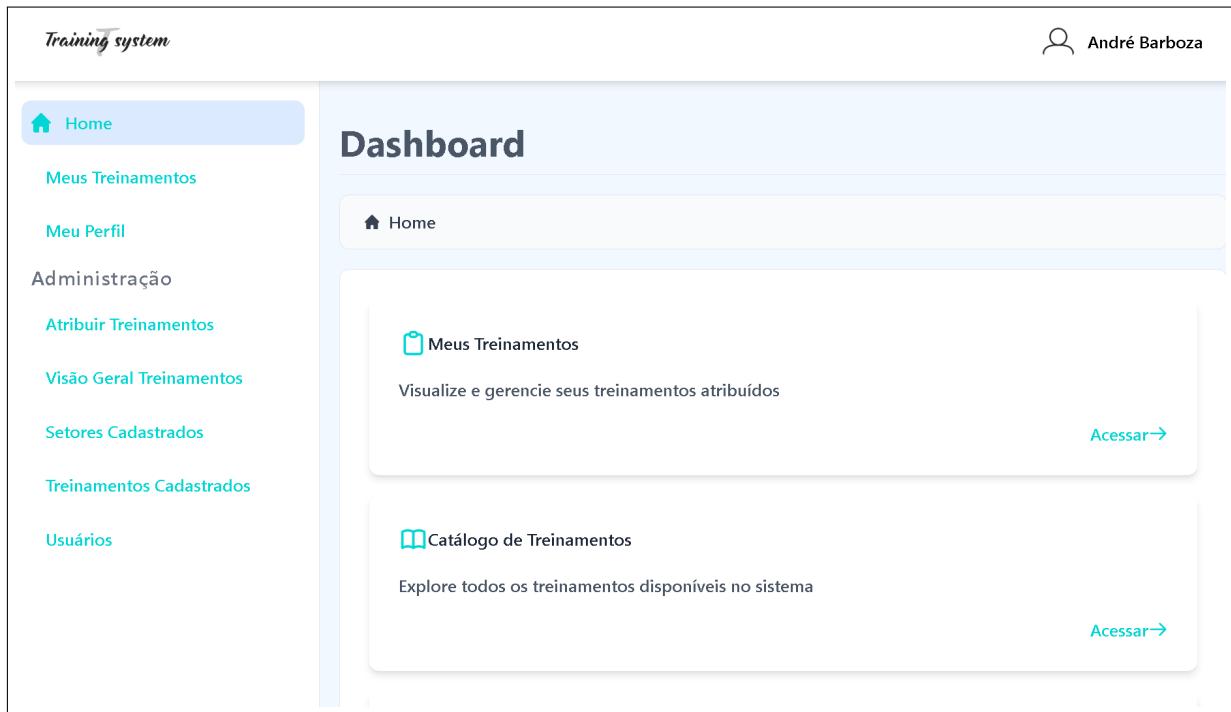


Figura 12 – Tela inicial do administrador.

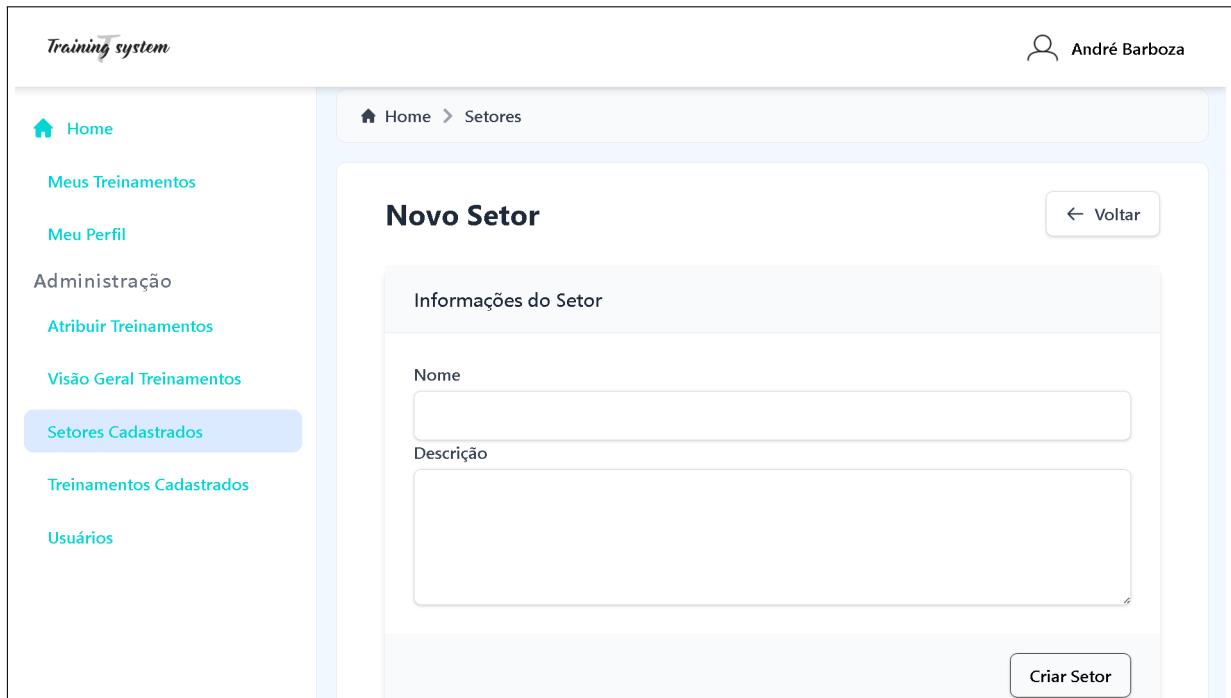


Figura 13 – Tela de cadastro de setor.

Figura 14 – Tela de cadastro de colaborador.

Figura 15 – Tela de vínculo de colaborador a setor.

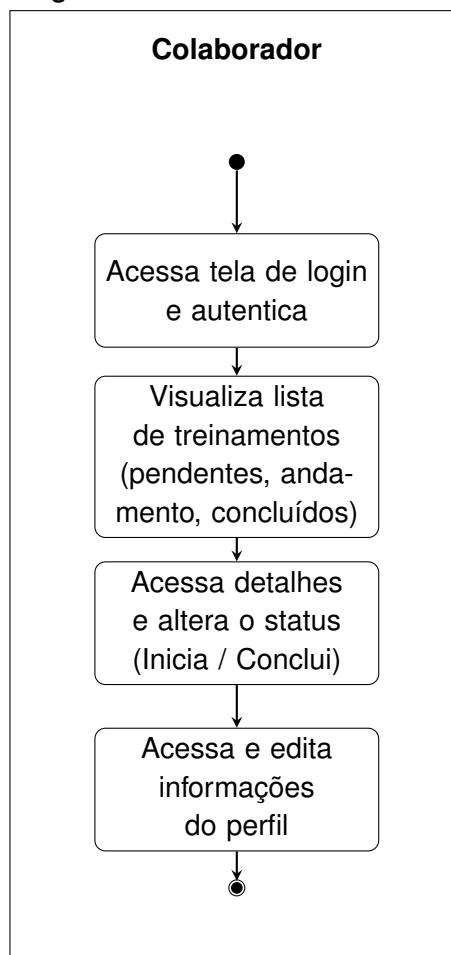
Figura 16 – Tela de cadastro de treinamento.

Figura 17 – Tela de monitoramento de status (Visão do Administrador).

#### 4.5.2 Fluxo do Colaborador

O colaborador utiliza o sistema principalmente para visualizar seus treinamentos, acompanhar seu progresso e gerenciar suas informações pessoais. O fluxo de atividades principal para este perfil, apresentado na Figura 18, foca na interação com os treinamentos atribuídos.

**Figura 18 – Diagrama de Atividades - Fluxo do Colaborador.**



As principais interações do colaborador, ilustradas com as telas da aplicação, são:

1. Ao logar, ele visualiza seus treinamentos, podendo ver quais estão pendentes, em andamento ou concluídos, como mostra a Figura 19.
2. O colaborador pode alterar o status de um treinamento atribuído a si, por exemplo, marcando-o como “Concluído” após sua realização, ação detalhada na Figura 20.
3. Ele também tem acesso ao seu perfil, onde pode visualizar e editar suas informações pessoais, conforme a Figura 21.

**Meus Treinamentos**

**Em Andamento** Treinamentos que você está realizando

Treinamento	Versão	Iniciado em	Ações
Treinamento teste	1	11 de outubro, 20:21	<a href="#">Detalhes</a>

**Concluídos** Treinamentos que você já completou

Treinamento	Versão	Concluído em	Válido até	Ações
Treinamento em ruby	2	16 de maio, 18:00	21 de maio	<a href="#">Detalhes</a>

**Figura 19 – Tela de listagem de treinamentos do colaborador.**

**11 de outubro de 2023, 20:21**

**Não concluído**

**Conteúdo do Treinamento**

Primeira versão do Treinamento teste onde será ensinado:

- Como criar treinamento
- Como ver treinamento

**Anexos**

home\_hero.png [Download](#)

**Concluir Treinamento**

**Figura 20 – Tela de detalhes do treinamento (em andamento) para o colaborador.**

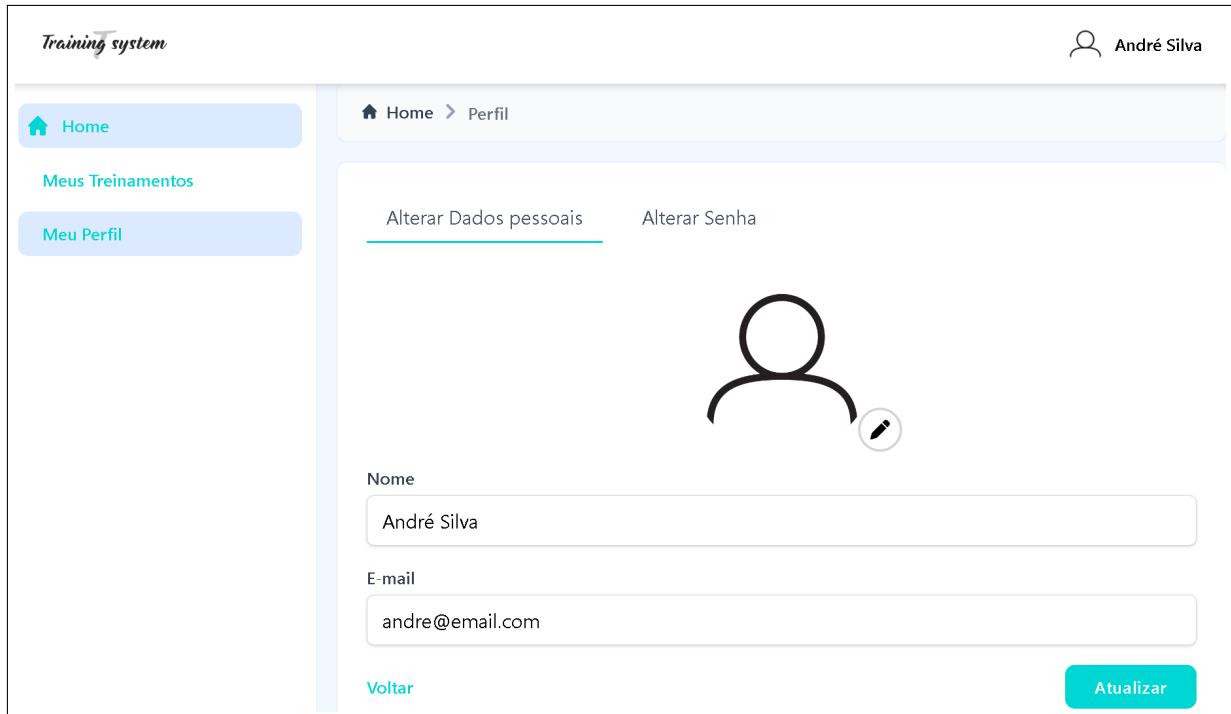


Figura 21 – Tela de perfil do colaborador.

## 4.6 Resultados Quantitativos

O desenvolvimento do Training System resultou em um sistema funcional que pode ser quantificado através das seguintes métricas de implementação e gestão de projeto.

### 4.6.1 Métricas de Implementação

A seguir serão apresentadas as características quantitativas resultantes do desenvolvimento do sistema:

- **Código-fonte Ruby/Rails:** 1.733 linhas de código distribuídas em componentes do framework
- **Código total (incluindo views):** 5.297 linhas, contemplando toda a lógica de negócio e interface
- **Arquivos do projeto:** 144 arquivos principais na camada de aplicação
- **Templates de interface:** 66 views (arquivos .erb) para renderização das páginas

#### 4.6.2 Estrutura da Aplicação (Padrão MVC)

A arquitetura do sistema seguiu o padrão MVC do Ruby on Rails, com a seguinte distribuição de componentes:

- **Models:** 10 models implementados, incluindo:
  - Entidades principais: User, Department, Training, TrainingVersion, UserTrainingVersion
  - Tabelas de associação: UserDepartment, DepartmentTraining
  - Models auxiliares: Session, Current, ApplicationRecord
  - Concern para funcionalidades compartilhadas: AccountConfirmable
- **Controllers:** 8 controllers principais desenvolvidos, responsáveis por gerenciar:
  - Autenticação e registro de usuários (Auth::SessionsController, Auth::PasswordsController, Auth::RegistrationsController)
  - Gestão de colaboradores (Users::UsersController, Users::ProfileController)
  - Gerenciamento de treinamentos (TrainingsController, TrainingVersionsController, UserTrainingVersionsController)
  - Administração de setores e vínculos (DepartmentsController, DepartmentTrainingsController)
- **Helpers:** 10 módulos auxiliares para suporte à renderização e formatação de dados nas Views
- **Mailers:** 3 classes para envio de e-mails (confirmação de conta, redefinição de senha e notificações)

#### 4.6.3 Atendimento aos Requisitos

Quanto ao cumprimento dos requisitos levantados nas Tabelas 3, 4 e 5:

- **Requisitos Must Have:** 18 requisitos essenciais implementados (100%)
  - 3 requisitos gerais (tela inicial, autenticação e guarda de rotas)
  - 4 requisitos para colaboradores
  - 11 requisitos para administradores
- **Requisitos Should Have:** 2 de 3 implementados

- Controle de versão dos treinamentos: implementado
- Padronização de documentos obrigatórios: implementado via automação de vínculos
- Notificações e alertas: implementado parcialmente (alertas visuais na interface)
- **Histórias de usuário:** 17 histórias de usuário concluídas e validadas (Apêndice A)
- **Interfaces desenvolvidas:** 66 templates de visualização, contemplando todas as funcionalidades de administrador e colaborador

#### 4.7 Resultados Qualitativos

Embora não tenham sido realizados testes de caso com métricas quantitativas, a comparação entre o fluxo de trabalho anterior (baseado em planilhas) e o *Training System* permite identificar resultados qualitativos significativos.

- **Agilidade no Acesso à Informação:** Observou-se uma redução significativa no tempo de consulta ao status de um treinamento. Antes, a verificação dependia de buscas manuais em planilhas, enquanto o novo sistema permite localizar colaboradores ou treinamentos de forma imediata.
- **Centralização e Integridade dos Dados:** O sistema eliminou a descentralização de informações. Ao contrário das planilhas, que eram vulneráveis a exclusões acidentais e não possuíam controle de versão, o sistema possibilita a integridade dos dados e registra o histórico de alterações, aumentando a confiabilidade das informações em auditorias.
- **Automatização de Processos:** A atribuição automática de treinamentos obrigatórios ao vincular um colaborador a um setor é um ganho de eficiência que reduz drasticamente o trabalho manual e o risco de falhas humanas.

## 5 CONSIDERAÇÕES FINAIS

A presente monografia detalhou o desenvolvimento do Training System, uma aplicação web concebida para solucionar as deficiências no controle de treinamentos de colaboradores, um problema real identificado no Capítulo 1 deste trabalho. O projeto partiu de um cenário baseado em planilhas e arquivos físicos, vulnerável a falhas e perdas de informação, e resultou em uma solução de software funcional e centralizada.

Para atingir este objetivo, o processo de desenvolvimento foi orientado por métodos ágeis, que precisaram ser adaptados à realidade de um projeto individual. Conforme descrito no Capítulo 2, o método MoSCoW foi fundamental para a priorização de requisitos, permitindo focar o escopo na entrega de um Produto Mínimo Viável (MVP) que atendesse às necessidades centrais do sistema, enquanto o Scrum e o GitFlow foram adaptados para um fluxo de trabalho individual.

As principais contribuições deste trabalho se materializam nas funcionalidades desenvolvidas, que atendem diretamente aos Objetivos específicos propostos. A implementação da automação na vinculação de treinamentos a setores (Seção 4.4.1), por exemplo, mitiga o risco de erro humano e o trabalho manual. Além disso, a criação de um sistema de controle de status (pendente, em andamento, concluído e expirado), com histórico de datas (Seção 4.4.2), atende ao objetivo de criar um registro rastreável e auditável, superando uma das principais limitações do método anterior (detalhado na seção 1.2) baseado em planilhas.

Dessa forma, o projeto cumpriu seu Objetivo Geral ao entregar uma ferramenta que oferece maior agilidade no acesso à informação, integridade dos dados e automação de processos, conforme demonstrado nos fluxos de atividades do Capítulo 4. O Training System representa, assim, uma solução estruturada para o gerenciamento de treinamentos, contribuindo para um melhor planejamento e controle de capacitações nas empresas.

Dentre as limitações metodológicas, a condução do desenvolvimento não seguiu integralmente as práticas do Scrum e do GitFlow, mas incorporou adaptações compatíveis com a execução individual e com restrições de tempo. O MoSCoW, por sua vez, foi aplicado integralmente e se mostrou decisivo para o direcionamento do trabalho. Assim, os métodos inicialmente planejados foram ajustados de acordo com a realidade do desenvolvimento, equilibrando a fundamentação teórica com a prática observada no projeto.

Os números apresentados nos resultados quantitativos (seção 4.6) demonstram que o sistema alcançou um nível adequado para atender aos objetivos propostos, com uma base de código estruturada e organizada seguindo as convenções do Framework Ruby on Rails.

## 5.1 Trabalhos Futuros

Embora o Training System atenda aos requisitos essenciais propostos, há um vasto campo para evoluções e novas funcionalidades. Como sugestões para trabalhos futuros, destacam-se:

- **Geração e Exportação de Relatórios:** Implementar um módulo que permita aos administradores gerar relatórios (ex: treinamentos vencidos, colaboradores por setor, etc.) e exportá-los em formatos como Formato de Documento Portátil, do inglês *Portable Document Format* (PDF) e Excel.
- **Dashboards Interativos:** Desenvolver painéis visuais com gráficos e *Key Performance Indicators* (KPIs), como a taxa de conclusão de treinamentos por equipe ou o percentual de conformidade da empresa.
- **Integração com Outros Sistemas:** Possibilitar a integração com sistemas de RH para sincronização de dados de colaboradores e com plataformas de e-learning para a emissão automática de certificados após a conclusão de um treinamento.

## REFERÊNCIAS

- AL1. **Pesquisa mostra que 94% das empresas brasileiras investem no treinamento de funcionários.** 2024. Acesso em: 23 out. 2024. Disponível em: <https://al1.com.br/informacao/noticias/131133/pesquisa-mostra-que-94-das-empresas-brasileiras-investem-no-treinamento-de-funcionarios>.
- Atlassian. **Três pilares do Scrum: compreensão dos princípios fundamentais do Scrum.** 2025. Acesso em: 12 fev. 2025. Disponível em: <https://www.atlassian.com/br/agile/project-management/3-pillars-scrum>.
- ClickUp. **ClickUp: One app to replace them all.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://www.clickup.com/>.
- Docker Inc. **Docker: Accelerate How You Build, Share, and Run Applications.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://www.docker.com/>.
- DRIESSEN, Vincent. **A successful Git branching model.** 2010. Acesso em: 12 fev. 2025. Disponível em: <https://nvie.com/posts/a-successful-git-branching-model/>.
- Figma. **Figma: The Collaborative Interface Design Tool.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://www.figma.com/>.
- GitHub. **GitHub: Where the World Builds Software.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://github.com/>.
- MkDocs. **MkDocs: Project documentation with Markdown.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://www.mkdocs.org/>.
- PX, T. **Git Flow Overview and Common Git Commands.** 2023. Acesso em: 11 fev. 2025. Disponível em: <https://dev.to/truongpx396/git-flow-overview-and-common-git-commands-1odh>.
- Ruby on Rails. **Ruby on Rails Guides.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://guides.rubyonrails.org/#main>.
- SEBRAE. **O método MoSCoW para definição de prioridades.** 2022. Acesso em: 12 fev. 2025. Disponível em: [https://sebrae.com.br/Sebrae/Portal%20Sebrae/Arquivos/ebook\\_sebrae\\_metodologia\\_moscow.pdf](https://sebrae.com.br/Sebrae/Portal%20Sebrae/Arquivos/ebook_sebrae_metodologia_moscow.pdf).
- SQLite. **SQLite: Database Engine.** 2024. Acesso em: 12 fev. 2025. Disponível em: <https://www.sqlite.org/>.
- SUTHERLAND, J. **Scrum: A Arte de Fazer o Dobro do Trabalho na Metade do Tempo.** Rio de Janeiro: Editora Sextante, 2014.

## **APÊNDICE A – Histórias de usuários e Protótipos de Tela**

Neste apêndice, são apresentadas as histórias de usuários e os protótipos das telas do sistema, que elucidam as funcionalidades principais do sistema.

### A.1 Histórias de usuários: Geral

Esta seção apresenta as histórias de usuário de escopo geral. Tais histórias contemplam as funcionalidades básicas do sistema, que são compartilhadas por ambos os perfis de usuário (administrador e colaborador). O objetivo é descrever a base de interação, como o acesso à tela inicial, a autenticação de usuário, o controle de rotas e o sistema de alertas.

#### Feature: Tela Inicial

**Como** usuário (administrador ou colaborador), **quero** visualizar tutoriais e manuais de uso **para** facilitar a navegação e o aprendizado sobre o sistema.

**Dado** que estou acessando o sistema,

**Quando** entro na página principal,

**Então** devo ver tutoriais e manuais de uso disponíveis.

#### Feature: Autenticação de usuário

**Como** usuário (administrador ou colaborador), **quero** autenticar-me no sistema **para** acessar as funcionalidades apropriadas ao meu perfil.

**Dado** que possuo credenciais válidas,

**Quando** faço login informando usuário e senha,

**Então** sou redirecionado para as funcionalidades específicas do meu perfil.

#### Feature: Guarda de rotas

**Como** usuário autenticado, **quero** que o sistema restrinja o acesso a áreas específicas **para** que somente usuários autorizados possam acessá-las.

**Dado** que estou logado como usuário,

**Quando** tento acessar uma área restrita,

**Então** o sistema verifica minhas permissões e, se não forem adequadas, nega o acesso.

#### Feature: Notificações e alertas

**Como** usuário (administrador ou colaborador), **quero** receber notificações e alertas sobre treinamentos pendentes, prazos e novas inscrições **para** não perder prazos e me manter atualizado.

**Dado** que existem treinamentos ou eventos relevantes,

**Quando** o sistema identifica que há um prazo próximo ou uma nova inscrição disponível,

**Então** devo receber um alerta ou notificação no sistema (ou via e-mail, caso implementado).

#### A.2 Histórias de usuários: : Usuário Administrador (*Must e Should*)

A área do usuário administrador permite que usuários com permissões específicas auxiliem na administração da plataforma, oferecendo suporte em atividades operacionais, monitorando a execução dos treinamentos de cada colaborador, controlando o progresso e assegurando a conformidade com os requisitos estabelecidos.

A seguir, são apresentadas as definições das telas do administrador, descritas em formato de histórias de usuário.

#### Feature: Registros de histórico

**Como** administrador, **quero** que todas as ações realizadas no sistema sejam registradas **para** fins de auditoria e acompanhamento de alterações.

**Dado** que estou autenticado como administrador,

**Quando** realizo alterações de status ou edito cadastros,

**Então** essas ações devem ser registradas em um histórico acessível somente para administradores.

#### Feature: Alteração de status do treinamento

**Como** administrador, **quero** modificar o status de um treinamento (pendente, em andamento, concluído) **para** manter o controle atualizado de cada etapa do processo.

**Dado** que existe um treinamento cadastrado,

**Quando** acesso o gerenciamento de treinamentos e selecione uma nova opção de status,

**Então** o sistema deve atualizar o status do treinamento e registrar essa mudança no histórico.

#### Feature: Cadastro de colaboradores

**Como** administrador, **quero** cadastrar novos colaboradores **para** que eles tenham acesso ao sistema e possam participar dos treinamentos.

**Dado** que sou um administrador autenticado,

**Quando** informo os dados do colaborador (nome, e-mail, cargo, etc.),

**Então** o colaborador deve ser registrado no sistema e receber acesso de acordo com seu perfil.

#### Feature: Cadastro de treinamentos

**Como** administrador, **quero** cadastrar novos treinamentos, definindo datas, instrutores e critérios de participação **para** organizar as ofertas de capacitação.

**Dado** que estou autenticado como administrador,

**Quando** insiro os dados necessários para um novo treinamento,

**Então** o sistema deve criar o registro do treinamento e disponibilizá-lo para os colaboradores.

#### Feature: Listagem de colaboradores

**Como** administrador, **quero** visualizar todos os colaboradores cadastrados **para** aplicar filtros e acompanhar o status de cada um.

**Dado** que sou um administrador autenticado,

**Quando** acesso a lista de colaboradores,

**Então** devo ver todos os colaboradores com a opção de filtrar e ordenar por nome, setor ou status.

#### Feature: Listagem de treinamentos (Administrador)

**Como** administrador, **quero** consultar todos os treinamentos cadastrados **para** acompanhar os detalhes, datas e instrutores.

**Dado** que existem treinamentos registrados,

**Quando** acesso a área de treinamentos como administrador,

**Então** devo poder visualizar a lista completa, com filtros por nome, data e instrutor.

#### Feature: Edição de colaboradores

**Como** administrador, **quero** atualizar os dados cadastrais de colaboradores **para** manter as informações corretas e atualizadas.

**Dado** que sou administrador,

**Quando** acesso o perfil de um colaborador e edito seus dados,

**Então** as alterações devem ser salvas e refletidas em todo o sistema.

#### Feature: Edição de treinamentos

**Como** administrador, **quero** editar as informações de um treinamento (datas, participantes, etc.) **para** ajustar detalhes conforme necessário.

**Dado** que já existe um treinamento cadastrado,

**Quando** acesso a opção de edição e altero os campos desejados,

**Então** o sistema deve atualizar os dados do treinamento e registrar essa mudança.

#### Feature: Exclusão de treinamentos

**Como** administrador, **quero** excluir treinamentos que não são mais relevantes **para** manter o sistema organizado e evitar sobrecarga de informações.

**Dado** que sou administrador,

**Quando** selecionei um treinamento para exclusão,

**Então** o sistema deve remover o registro, desde que não afete dados críticos, e registrar essa ação no histórico.

#### Feature: Exclusão de colaboradores

**Como** administrador, **quero** remover colaboradores inativos **para** manter o cadastro atualizado, preservando o histórico de treinamentos realizados.

**Dado** que um colaborador está inativo,

**Quando** selecionei a opção de exclusão,

**Então** o sistema deve desativar o colaborador, mantendo os registros históricos associados.

#### Feature: Controle de acesso

**Como** administrador, **quero** definir perfis de acesso **para** garantir que cada usuário tenha permissões adequadas às suas funções.

**Dado** que possuo privilégios de administrador,

**Quando** gerencio as configurações de perfis,

**Então** posso atribuir e remover permissões de acordo com as necessidades do sistema.

#### Feature: Perfil do administrador

**Como** administrador, **quero** visualizar e editar minhas próprias informações **para** manter meus dados atualizados e gerenciar contas de outros usuários.

**Dado** que estou autenticado como administrador,

**Quando** acesso a seção de perfil,

**Então** posso alterar meus dados pessoais e gerenciar as contas de outros usuários.

#### Feature: Controle de versão dos treinamentos

**Como** administrador, **quero** registrar e atualizar as versões dos treinamentos **para** garantir que os colaboradores sempre tenham acesso ao conteúdo mais recente.

**Dado** que possuo um treinamento já cadastrado,

**Quando** faço uma atualização de conteúdo,

**Então** o sistema deve criar uma nova versão e disponibilizá-la para os colaboradores, mantendo o histórico de versões anteriores.

#### Feature: Padronização de documentos obrigatórios

**Como** administrador, **quero** automatizar a atribuição de treinamentos obrigatórios a novos colaboradores **para** garantir que todos recebam o mesmo nível de formação inicial.

**Dado** que um novo colaborador é cadastrado,

**Quando** o sistema identifica que existem treinamentos obrigatórios,

**Então** esses treinamentos devem ser vinculados automaticamente ao colaborador.

### A.3 Histórias de usuários: Usuário Colaborador (*Must* e *Should*)

Para o perfil de colaborador, as histórias de usuário demonstram as funcionalidades essenciais que possibilitam o acesso aos treinamentos, a visualização de informações detalhadas e a gestão dos dados pessoais.

#### Feature: Listagem de treinamentos (Colaborador)

**Como** colaborador, **quero** visualizar todos os treinamentos disponíveis **para** acompanhar meu progresso e identificar quais ainda estão pendentes, em andamento ou concluídos.

**Dado** que estou autenticado como colaborador,

**Quando** acesso a lista de treinamentos,

**Então** devo ver os treinamentos com status e detalhes relevantes.

#### Feature: Detalhes do treinamento

**Como** colaborador, **quero** visualizar informações completas de um treinamento **para** compreender sua descrição, datas, instrutor responsável e status.

**Dado** que existe um treinamento cadastrado,

**Quando** seleciono um treinamento específico,

**Então** devo ver todos os detalhes, incluindo a descrição, datas, instrutor e status.

#### Feature: Perfil do colaborador

**Como** colaborador, **quero** visualizar e editar minhas informações básicas **para** manter meus dados atualizados (nome, setor e cargo).

**Dado** que estou autenticado como colaborador,

**Quando** acesso meu perfil,

**Então** posso editar minhas informações e salvar as alterações.

#### Feature: Vínculo de colaborador e treinamento

**Como** colaborador, **quero** ser associado aos treinamentos obrigatórios **para** garantir que os documentos necessários sejam registrados e que eu possa participar das capacitações.

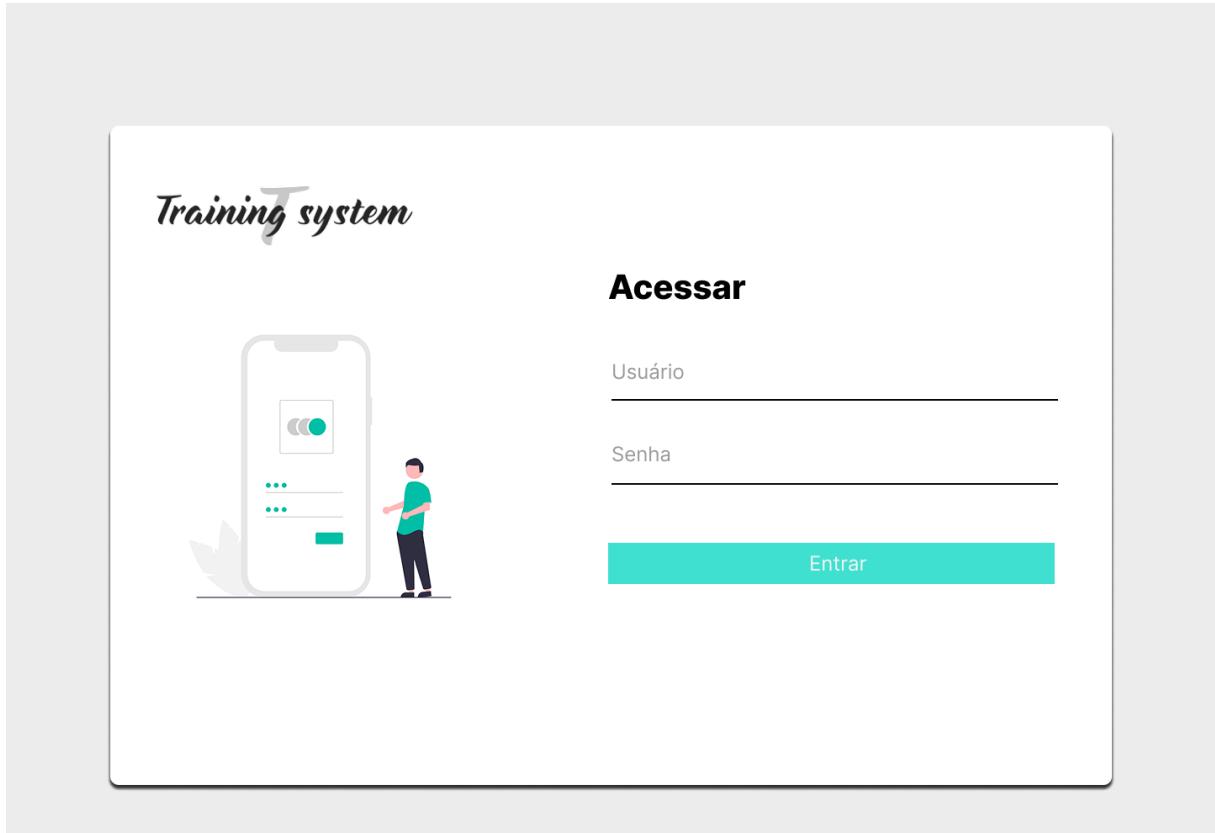
**Dado** que fui cadastrado no sistema,

**Quando** um treinamento obrigatório é definido,

**Então** devo ser vinculado a esse treinamento e ter acesso aos documentos associados.

#### A.4 Protótipo de Telas: Área do Colaborador

Nesta subseção, apresenta-se a interface destinada aos colaboradores. O acesso inicial se dá por meio da tela de login, conforme exibido na Figura 22, onde o usuário informa suas credenciais para autenticação.



**Figura 22 – Tela de Login: interface de autenticação do usuário. Fonte: Autoria própria.**

A Figura 23 exibe a tela de listagem de treinamentos, onde são apresentados o nome, a categoria, o prazo de validade e as opções de edição ou remoção dos registros.

Em seguida, a Figura 24 apresenta a tela de ajustes de conta, que oferece opções de configuração de perfil acessíveis tanto para administradores quanto para colaboradores.

#### A.5 Protótipo de Telas: Área do Administrador

Nesta subseção, são detalhadas as interfaces destinadas ao administrador. Após realizar o login (inicialmente como colaborador), o administrador é redirecionado para a tela inicial do sistema, demonstrada na Figura 25.

A Figura 26 apresenta a tela de cadastro de treinamentos, que permite ao administrador registrar novos treinamentos no sistema.

Em seguida, na Figura 27, exibe-se a tela de cadastro de colaboradores, destinada ao registro de novos usuários.



Olá, André!  
Opções Conta >

Cadastro Treinamento Listagem Treinamento Cadastro Colaborador Listagem Colaborador Vincular

**Listagem de Treinamentos**

Prazo de Validade	Treinamento	Categoria
12/03/2029	Relacionamento Banco de Dados	Manutenção
12/03/2029	Relacionamento Banco de Dados	Produção
12/03/2029	Relacionamento Banco de Dados	Administrativo
12/03/2029	Relacionamento Banco de Dados	Terceiros
12/03/2029	Relacionamento Banco de Dados	Novo colaborador
12/03/2029	Relacionamento Banco de Dados	Infraestrutura
12/03/2029	Relacionamento Banco de Dados	Expedição

[Editar](#) [Remover](#)

Versão 0.1  
Data de Atualização 01/08/2023

(42)3623-5987 suporte@training.com

**Figura 23 – Listagem de Treinamentos: visualização dos treinamentos disponíveis. Fonte: Autoria própria.**



Alterar Imagem

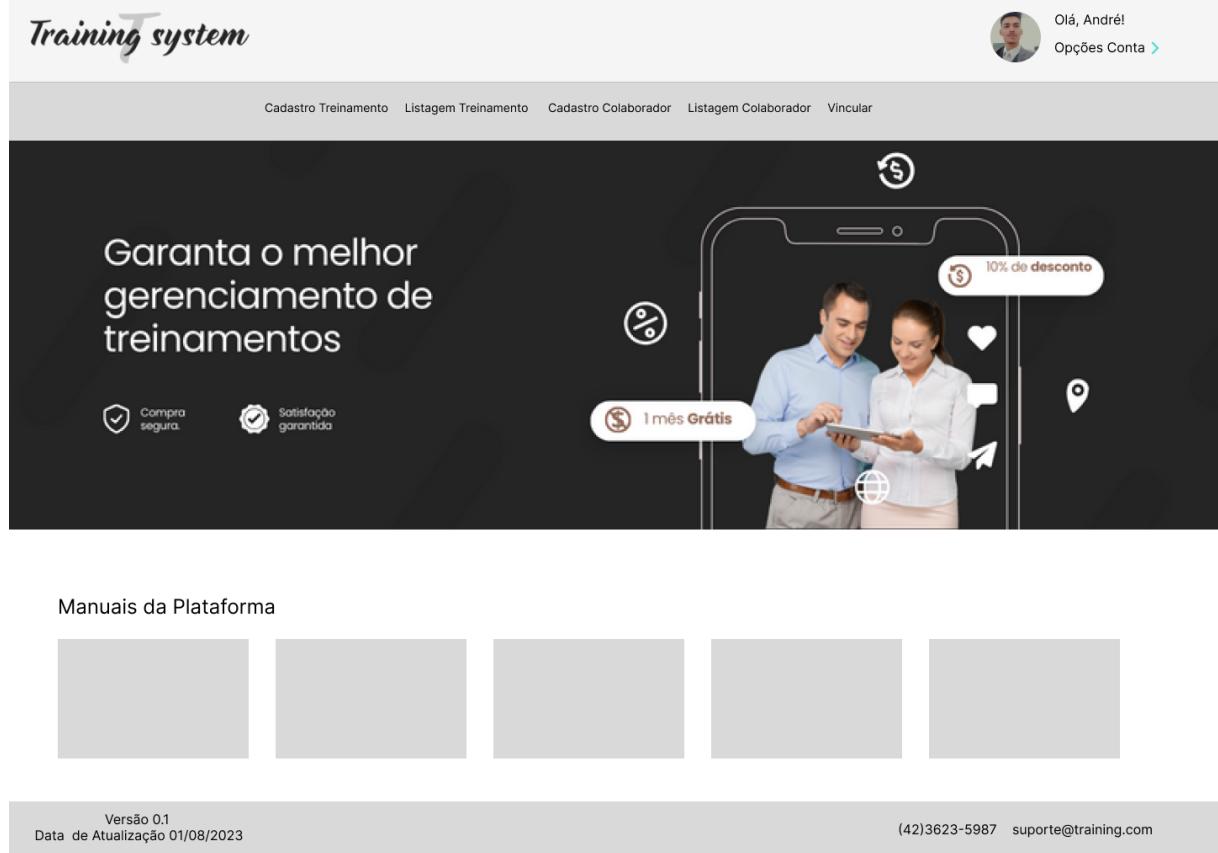
Nome

Alterar senha

Confirmar senha

[Salvar](#) [Cancelar](#)

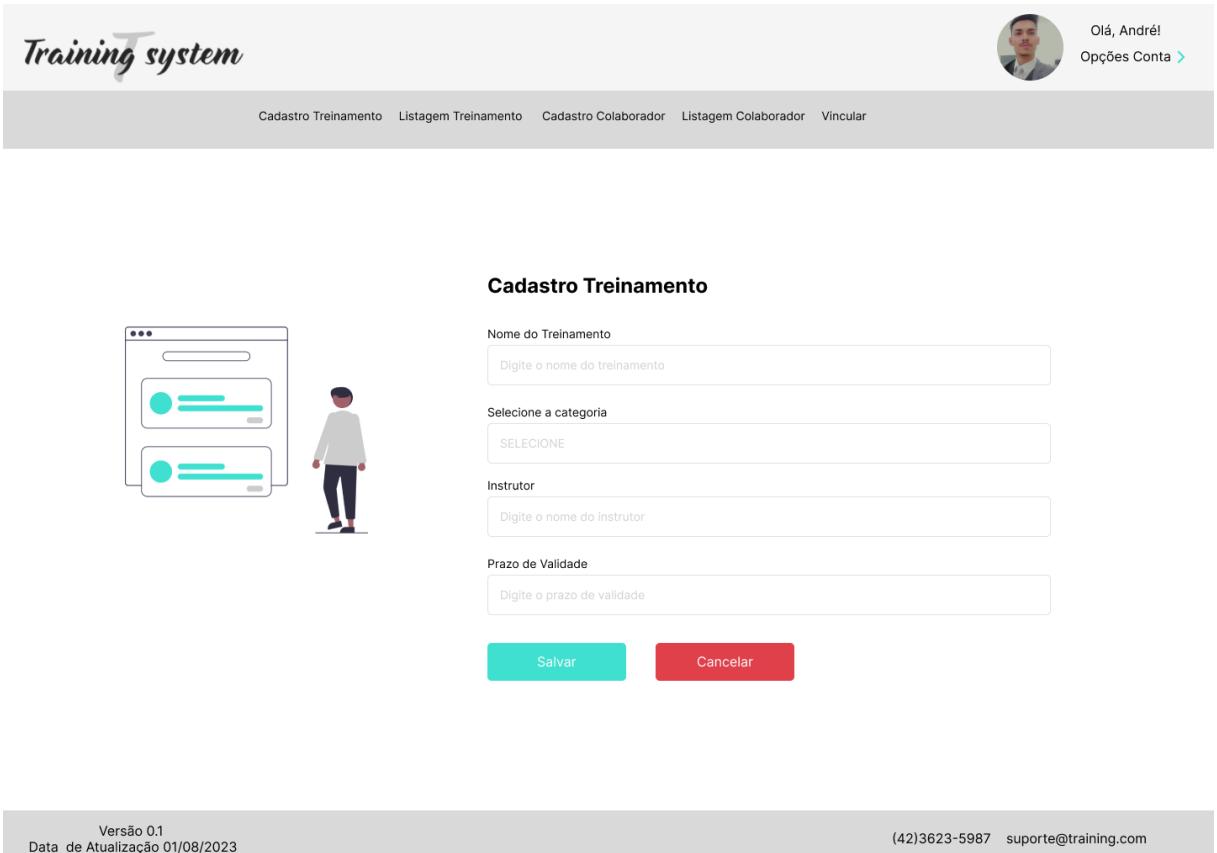
**Figura 24 – Opções de Conta: configurações de perfil e ajustes pessoais. Fonte: Autoria própria.**



**Figura 25 – Tela Inicial do Administrador: painel principal de acesso e navegação. Fonte: Autoria própria.**

A Figura 28 mostra a tela de listagem de colaboradores, que possibilita a visualização e o gerenciamento dos registros dos usuários cadastrados.

Por fim, a Figura 29 ilustra a interface de vinculação de funcionários a treinamentos, permitindo a associação dos colaboradores aos treinamentos correspondentes.



The screenshot shows the 'Training system' application interface. At the top, there is a navigation bar with links: 'Cadastro Treinamento', 'Listagem Treinamento', 'Cadastro Colaborador', 'Listagem Colaborador', and 'Vincular'. On the right side of the header, there is a user profile placeholder with the text 'Olá, André!' and 'Opções Conta >'. The main content area is titled 'Cadastro Treinamento'. It features a form with the following fields: 'Nome do Treinamento' (input placeholder: 'Digite o nome do treinamento'), 'Selecione a categoria' (input placeholder: 'SELECIONE'), 'Instrutor' (input placeholder: 'Digite o nome do instrutor'), and 'Prazo de Validade' (input placeholder: 'Digite o prazo de validade'). Below the form are two buttons: 'Salvar' (Save) in a teal box and 'Cancelar' (Cancel) in a red box. At the bottom of the page, there is a footer with the text 'Versão 0.1' and 'Data de Atualização 01/08/2023' on the left, and '(42)3623-5987 suporte@training.com' on the right.

**Figura 26 – Cadastro de Treinamentos: formulário para inclusão de novos treinamentos. Fonte: Autoria própria.**

The screenshot shows the 'Training system' interface. At the top, there is a navigation bar with links: 'Cadastro Treinamento', 'Listagem Treinamento', 'Cadastro Colaborador', 'Listagem Colaborador', and 'Vincular'. On the right side of the header, there is a user profile placeholder with the text 'Olá, André!' and 'Opções Conta >'. Below the header, the main content area is titled 'Cadastro Colaborador'. On the left, there is a small illustration of a person standing next to a computer monitor displaying a user profile. The form fields are as follows:

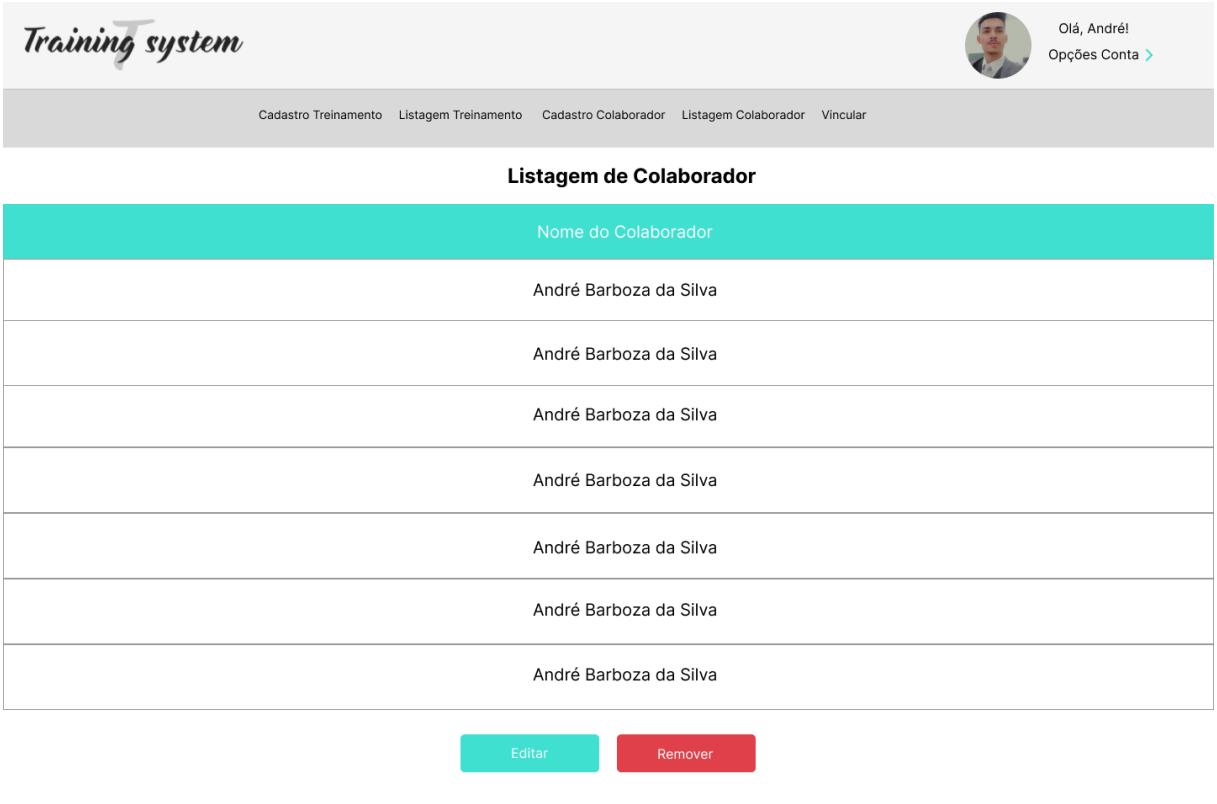
- Usuário de acesso:** Input field with placeholder 'Digite o usuário de acesso do colaborador'.
- Nome do Colaborador:** Input field with placeholder 'Digite o nome do colaborador'.
- Senha:** Input field with placeholder 'Digite a senha'.
- Confirmar Senha:** Input field with placeholder 'Confirme a senha'.

At the bottom of the form are two buttons: 'Salvar' (Save) in a teal box and 'Cancelar' (Cancel) in a red box.

**Versão 0.1**  
Data de Atualização 01/08/2023

(42)3623-5987 suporte@training.com

**Figura 27 – Cadastro de Colaboradores: interface para inclusão de novos usuários. Fonte: Autoria própria.**



Training system

Olá, André! [Opções Conta >](#)

Cadastro Treinamento Listagem Treinamento Cadastro Colaborador Listagem Colaborador Vincular

**Listagem de Colaborador**

Nome do Colaborador
André Barboza da Silva

[Editar](#) [Remover](#)

Versão 0.1  
Data de Atualização 01/08/2023

(42)3623-5987 [suporte@training.com](mailto:suporte@training.com)

**Figura 28 – Listagem de Colaboradores: visualização dos registros e gerenciamento dos usuários. Fonte: Autoria própria.**

The screenshot shows the 'Training system' interface. At the top right, there is a user profile with the name 'Olá, André!' and a 'Opções Conta' link. The main title is 'Vincular colaborador'. Below the title, there are two input fields: 'Nome do Colaborador' and 'Nome do Treinamento', both with placeholder text 'Selecione o colaborador' and 'Selecione o treinamento'. A large green 'Vincular' button is centered below these fields. At the bottom, a table lists three associations, each with a yellow status circle, the name 'André Barboza da Silva', the category 'Relacionamento Banco de Dados', and the status 'Pendente'. The table includes a red trash icon for each row. At the very bottom, there is a footer with the text 'Versão 0.1' and 'Data de Atualização 01/08/2023' on the left, and '(42)3623-5987 suporte@training.com' on the right.

André Barboza da Silva	Relacionamento Banco de Dados	<span style="color: yellow;">●</span> Pendente	<span style="color: red;">●</span>
André Barboza da Silva	Relacionamento Banco de Dados	<span style="color: yellow;">●</span> Pendente	<span style="color: red;">●</span>
André Barboza da Silva	Relacionamento Banco de Dados	<span style="color: yellow;">●</span> Pendente	<span style="color: red;">●</span>

**Figura 29 – Vinculação de Funcionários: interface para associação de colaboradores aos treinamentos. Fonte: Autoria própria.**